



**Sílvia Isabel Moreira
Monteiro**

**Sistema automático para correção de artefactos
oculares com SVD**



**Sílvia Isabel Moreira
Monteiro**

**Sistema automático para correção de artefactos
oculares com SVD**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica de Ana Maria Perfeito Tomé, Professora do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Professor Doutor Tomás António Mendes Oliveira e Silva
Professor Associado da Universidade de Aveiro

vogais / examiners committee

Professora Doutora Ana Maria Perfeito Tomé
Professora Associada da Universidade de Aveiro (orientadora)

Professora Doutora Pétia Georgieva
Professora Auxiliar da Universidade de Aveiro (co-orientadora)

Professor Doutor Aníbal João de Sousa Ferreira
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

**agradecimentos /
acknowledgements**

Em primeiro lugar agradeço à minha orientadora Professora Doutora Ana Maria Tomé por toda a dedicação e disponibilidade demonstradas ao longo deste trabalho.

Em segundo lugar agradeço à Psicóloga Mestre Ana Teresa Pereira pela colaboração prestada neste trabalho.

Um agradecimento especial à minha família pois sem eles teria sido impossível ter chegado até aqui e ao meu namorado por todo o apoio dado durante esta etapa.

E por fim, agradeço a todos os meus amigos e colegas de curso pela ajuda e companheirismo demonstrado ao longo destes anos.

Resumo

O eletroencefalograma (EEG) é atualmente uma técnica de diagnóstico médico, não invasiva e pouco dispendiosa, que mede a função do cérebro, através da monitorização da actividade elétrica gerada pelas estruturas cerebrais. As técnicas de processamento dos eletroencefalogramas são, contudo, complexas. Os padrões obtidos nos registos variam com o sujeito e o seu estado de espírito, e podem ser afetados por diversos tipos de perturbações, como por exemplo movimentos oculares.

Este trabalho vem apresentar um sistema de correção automática de artefactos causados por movimentos oculares, baseado na decomposição em valores singulares (SVD). Foram desenvolvidas duas técnicas (PCA e SSA) que exploram correlações entre canais e entre amostras, respetivamente. Os coeficientes de correlação e o espectrograma são duas ferramentas que têm um papel importante na validação das técnicas implementadas.

Pretende-se assim, com este trabalho, estudar técnicas baseadas na decomposição em valores singulares (SVD) para correção de artefactos oculares.

Abstract

The Electroencephalogram EEG is currently a medical diagnosis technique, noninvasive and inexpensive, which measures the brain function through electrical activity generated by brain structures. The processing techniques of electroencephalograms are complex, however. The patterns obtained in the records change with the subject and his/her state of mind, and it can be affected by various types of disturbances, for example eye movements.

This work presents a system for correcting ocular artifacts based on singular value decomposition (SVD). Two techniques have been developed (PCA and SSA) that exploit correlations between channels and between samples, respectively. The correlation coefficients and the spectrogram are two tools that have an important role to validate the techniques implemented.

This work aims to study techniques based on singular value decomposition (SVD) for correction of ocular artifacts .

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Acrónimos	vii
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	1
1.3 Estrutura da Tese	2
2 Artefatos e EEG	3
2.1 Introdução	3
2.2 Potenciais evocados	3
2.3 Aquisição do sinal EEG	5
2.4 Processamento de Potenciais Evocados	6
2.5 Regressão linear na correção de artefatos	8
2.6 Outros métodos de correção de artefactos oculares	10
2.7 Conclusões	10
3 SVD e remoção de artefatos	11
3.1 Introdução	11
3.1.1 Análise multicanal - PCA	11
3.1.2 Análise canal a canal - SSA	13
3.2 Conclusões	18
4 O projecto do sistema automático	19
4.1 Introdução	19
4.2 Descrição global do sistema	19
4.3 Interface gráfica	19
4.3.1 Software da interface desenvolvido	21
4.4 Módulo de Processamento	21
4.4.1 Detetor de movimentos oculares	22
4.4.2 Restante software desenvolvido	24
4.5 Conclusões	26

5	Resultados	27
5.1	Introdução	27
5.2	Aplicação das técnicas de correção	27
5.3	Características do Registo 1	28
5.3.1	Análise no tempo das técnicas implementadas	29
5.3.2	Análise tempo - frequência das técnicas implementadas	32
5.4	Características do Registo 2	36
6	Conclusões e trabalho futuro	43
6.1	Discussão geral dos resultados	43
6.2	Trabalho futuro	44
	Apêndices	45
A	Exemplo de uma estrutura criada pela EEGLab	47
B	Código Matlab desenvolvido	49
B.1	Interface gráfica	49
B.2	Detector de picos	57
B.3	Função principal	58
B.4	Método Regressivo	59
B.5	Método PCA	60
B.6	Método SSA	60
	Bibliografia	63

Lista de Figuras

2.1	Dispositivos utilizados para avaliação da valência - 2.1a - e da arousal - 2.1b dos estímulos visuais - imagem adaptada de [1]	4
2.2	Representação da disposição de elétrodos segundo o sistema 10-20 (imagem adaptada de [5])	5
2.3	Posições dos eletrodos para aquisição dos sinais oculares (imagem adaptada de [4])	6
2.4	Esquema de apresentação dos estímulos no registo das medidas - imagem adaptada de [1]	6
2.5	Vários passos por que passa o ficheiro contínuo registado inicialmente	7
2.6	Passos para tratamento do sinal EEG: (a) apresenta o sinal contínuo sem qualquer tipo de tratamento; (b) apresenta o sinal depois de filtrado (0.1Hz - 30Hz); (c) apresenta o sinal depois de corrigido artefacto ocular; (d) apresenta o sinal depois de dividido em épocas e de corrigida linha de base	7
2.7	Efeito do movimento ocular no sinal de EEG, mais propriamente no canal Fz (imagem adaptada de [4])	8
2.8	Canal Fz contaminado (3), corrigido pelo método regressivo (1) e corrigido pelo Neuroscan (2)	9
3.1	Sinal de EEG utilizado (10 - 15 segundos)	12
3.2	Método multicanal aplicado a um pestanejo (400ms): (a) - Pestanejo marcado na figura 3.1; (b) - Pestanejo depois de transformado em componentes principais; (c) - Rejeição de artefacto aplicado às duas primeiras componentes; (d) - Correção de artefacto aplicado às duas primeiras componentes	12
3.3	(a) - Pestanejo marcado na figura 3.1 ; (b) - Pestanejo depois de rejeitado artefacto ocular; (c) - Pestanejo depois de corrigido artefacto ocular	13
3.4	Canal a tratar e respectivos atrasos temporais (400ms): (a) - Pestanejo marcado na figura 3.1 (Canal FP1); (b) - M=10 - Sinal depois de atrasado 10 vezes (convertido em 10 componentes); (c) - M=20 - Sinal depois de atrasado 20 vezes (convertido em 20 componentes); (d) - M=50 - Sinal depois de atrasado 50 vezes (convertido em 50 componentes)	15
3.5	Aplicação do método com rejeição do artefacto: (a) M=10; (b) M=20; (c) M=50	16
3.6	Aplicação do método com correção do artefacto à matriz \mathbf{X} : (a) M=10; (b) M=20; (c) M=50.	16
3.7	Sinal recuperado depois de aplicação do método com opção de rejeição do artefacto: (a) M=10; (b) M=20; (c) M=50.	17

3.8	Sinal recuperado depois de aplicação do método com opção de correcção do artefacto: (a) M=10; (b) M=20; (c) M=50.	17
4.1	Diagrama de blocos geral do sistema	19
4.2	Diagrama de blocos da interface gráfica desenvolvida	20
4.3	Interface gráfica desenvolvida	20
4.4	Diagrama de blocos do módulo de processamento	22
4.5	(a) Detector de picos (máximos e mínimos) ; (b) Movimentos oculares apenas com a detecção dos máximos.	23
5.1	Canal FP1 e Canal VEOG (10 -11 segundos)	28
5.2	Caixa de bigodes dos coeficientes de correlação entre o sinal contaminado e o canal ocular VEOG calculados nos 400ms de cada pestanejo	28
5.3	Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo MR: (a) com o canal ocular VEOG e (b) com o sinal contaminado , calculados nos 400ms de cada pestanejo.	29
5.4	Comparação de sinais com canal FP1 depois de aplicado MR(10 -11 segundos)	29
5.5	Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método: PCA - CORR: (a) com o canal ocular VEOG e (b) com o sinal contaminado; PCA - EL: (c) com o canal ocular VEOG e (d) com o sinal contaminado.	30
5.6	Comparação de sinais com canal FP2 depois de aplicado PCA - CORR(10 -11 segundos)	31
5.7	Comparação de sinais com canal FPZ depois de aplicado SSA - CORR com 10 componentes (10 -11 segundos)	31
5.8	Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método (SSA - CORR), calculados nos 400ms de cada pestanejo: Correlação com sinal ocular - (a) M=10 , (c) M=20 e (e) M=50; Correlação com sinal contaminado - (b) M=10 , (d) M=20 e (f) M=50.	32
5.9	Sinal de EEG (canal FP1) entre 10 e 15 segundos: (a) Sinal contaminado, (b) Análise na frequência do sinal contaminado, (c) Sinal corrigido pelo MR, (d) Análise na frequência do sinal corrigido pelo MR.	33
5.10	Sinal de EEG (canal FP2) entre 10 e 15 segundos: (a) Sinal contaminado, (b) Análise na frequência do sinal contaminado, (c) Sinal corrigido pelo método PCA - CORR, (d) Análise na frequência do sinal corrigido pelo método PCA - CORR, (e) Sinal corrigido pelo método PCA - EL, (f) Análise na frequência do sinal corrigido pelo método PCA - EL.	34
5.11	Sinal de EEG (canal FPZ) entre 10 e 15 segundos: (a) Sinal contaminado, (b) Análise na frequência do sinal contaminado, (c) Sinal corrigido pelo método SSA - CORR com M=10, (d) Análise na frequência do sinal corrigido pelo método SSA - CORR com M=10, (e) Sinal corrigido pelo método SSA - EL com M=10, (f) Análise na frequência do sinal corrigido pelo método SSA - EL com M=10.	35
5.12	(a) Sinal de EEG contaminado e (b) depois de corrigido pelo método PCA - CORR	36
5.13	Caixa de bigodes dos coeficientes de correlação entre o sinal contaminado e o canal ocular VEOG calculados nos 400ms de cada pestanejo	37
5.14	Canal FP1 e Canal VEOG (11 -13 segundos)	37

5.15	Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo MR: (a) com o canal ocular VEOG e (b) com o sinal contaminado, calculados nos 400ms de cada pestanejo	38
5.16	Comparação de sinais com canal FP1 depois de aplicado MR(11 -13 segundos) .	38
5.17	Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método: PCA - CORR: (a) com o canal ocular VEOG e (b) com o sinal contaminado; PCA - EL: (c) com o canal ocular VEOG e (d) com o sinal contaminado.	39
5.18	Comparação de sinais com canal FP2 depois de aplicado PCA - CORR(11 -13 segundos)	40
5.19	Comparação de sinais com canal FPZ depois de aplicado método SSA - CORR com 10 componentes (11 -13 segundos)	40
5.20	Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método SSA - CORR, calculados nos 400ms de cada pestanejo: Correlação com sinal ocular e (a) M=10, (c) M=20 e (e) M=50; Correlação com sinal contaminado e (b) M=10 , (d) M=20 e (f) M=50.	41
A.1	Exemplo de uma estrutura de dados do EEGLab, depois de carregado ficheiro contínuo	47

Acrónimos

EEG Eletroencefalograma

EOG Eletrooculograma

HEOG Eletrooculograma Horizontal

ICA Análise de componentes independentes

MR Método Regressivo

PCA Análise de componentes principais

PCA - EL Análise de componentes principais com eliminação de componentes

PCA - CORR Análise de componentes principais com correção de componentes

SSA Análise espectral simples

SVD Decomposição em valores singulares

SSA - EL Análise espectral simples com eliminação de componentes

SSA - CORR Análise espectral simples com correção de componentes

VEOG Eletrooculograma Vertical

Capítulo 1

Introdução

1.1 Enquadramento

O eletroencefalograma EEG é um registo dos potenciais elétricos no couro cabeludo provenientes da atividade eletrofisiológica do cérebro. Atualmente estes sinais além de serem utilizados em diagnósticos clínicos, também são utilizados em interfaces humano-computador. O registo dos sinais de EEG pode sofrer interferências de outros sinais como, por exemplo o pestanejar dos olhos.

O olho forma um dipolo elétrico onde a córnea é positiva e a retina é negativa. Quando os olhos se movem, o campo elétrico à volta dos olhos altera-se, produzindo um sinal elétrico conhecido como o Electrooculograma (EOG). Como este sinal se propaga sobre o escalpe, vai aparecer no sinal de EEG adquirido como ruído ou artefactos oculares que vão dificultar a interpretação e análise deste. Este sinal interfere, principalmente, nos canais frontais. Surge assim a necessidade de criar técnicas de correção de artefactos oculares.

A correção de artefactos oculares tem sido um dos problemas mais estudados no processamento de sinal EEG. As técnicas baseadas em regressão linear foram as primeiras a surgir, nomeadamente regressão linear simples no domínio do tempo [17] [7] e regressão linear no domínio da frequência [19] [20]. Em 2000, surge uma técnica baseada na análise de componentes independentes (ICA) [11] que separa os sinais EOG dos sinais EEG. Esta técnica requer uma análise e processamento “offline” dos dados recolhidos a partir de um número elevado de canais, e o seu sucesso depende de uma correta identificação das componentes com ruído. Estas duas técnicas (Regressão linear e ICA) são utilizadas pelo Neuroscan e EEGLab, respetivamente, para correção de artefactos oculares.

1.2 Objetivos

Este trabalho teve como objetivo estudar técnicas baseadas na decomposição em valores singulares (SVD) para correção de artefactos oculares. A decomposição SVD é utilizada em duas variantes em que se manipulam todos os canais do EEG ou apenas um deles. Os coeficientes de correlação e o espectrograma são duas ferramentas que vão ter um papel importante para validar as técnicas implementadas. Para automatizar a correção recorreu-se a programação Matlab e, em particular, às funções para criação de interface gráfica.

1.3 Estrutura da Tese

Este trabalho descreve todo o trabalho realizado para a avaliação de métodos baseados em SVD. A estrutura deste encontra-se dividida em 6 capítulos, incluindo o presente (Capítulo 1) onde se faz uma introdução ao trabalho e se apresenta a estrutura.

O Capítulo 2 descreve o que são artefactos oculares e de que forma estes podem afetar os sinais de EEG. É apresentado também o protocolo seguido para aquisição dos sinais de EEG utilizados para análise do sistema, e onde indica os diferentes tipos de estímulos existentes. A terminar o capítulo é explicado de que forma a regressão linear atua na correção de artefactos oculares, acabando com algumas conclusões deste capítulo.

Toda a descrição do que consistem as técnicas implementadas neste trabalho está presente no Capítulo 3. É neste capítulo que são explicadas em pormenor as técnicas PCA e SSA e de que forma elas podem melhorar a tarefa de correção de artefactos oculares presentes em sinais EEG.

O capítulo seguinte (Capítulo 4) faz uma descrição do projeto do sistema automático desenvolvido, fazendo uma divisão entre interface gráfica e módulo de processamento. Em ambos os módulos encontram-se explicadas todas as funções utilizadas. É de notar que o sistema foi desenvolvido em programação Matlab.

Os dois últimos capítulos apresentam todos os resultados obtidos (Capítulo 5) e as conclusões finais do trabalho (Capítulo 6). Neste último são ainda sugeridas propostas de trabalho a realizar no futuro.

Capítulo 2

Artefatos e EEG

2.1 Introdução

O eletroencefalograma (EEG) consiste num processo de registo da atividade elétrica do cérebro, produzida nos neurónios corticais, e cuja tensão pode variar entre uma amplitude de 2 a várias centenas de micro-Volt. A aquisição de um sinal contínuo de EEG pode ser severamente contaminada por artefactos fisiológicos, tais como movimentos oculares (EOG) e atividade cardíaca ou muscular [8].

Com a disponibilização do sinal de EEG digital, tornou-se desejável e necessário tentar encontrar um procedimento capaz de corrigir um artefacto presente no sinal adquirido de forma a não perder informação.

2.2 Potenciais evocados

Os potenciais evocados são sinais elétricos gerados pelo sistema nervoso em resposta a estímulos sensoriais [15], e são medidos através do EEG. O objetivo principal destes é estudar o funcionamento, e analisar o grau de integridade, dos caminhos neuro-fisiológicos percorridos pelas correntes eletroquímicas (das quais o efeito medido são potenciais elétricos), ao longo dos diversos nervos até ao cérebro. Esses potenciais produzidos constituem a resposta a um ou mais estímulos. Existem três tipos de estímulos:

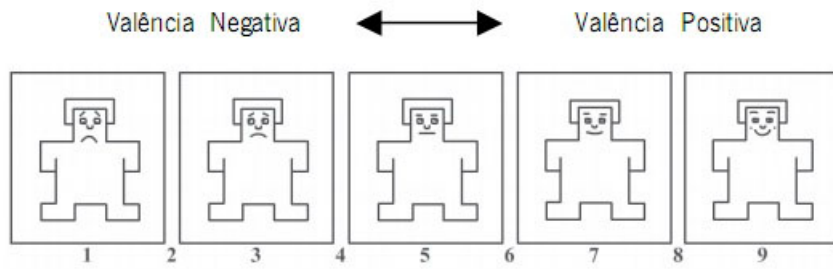
- Estímulos Sensoriais
- Estímulos Auditivos
- Estímulos Visuais

Para além desta divisão, os estímulos podem ser classificados como potenciais de curta, média e longa latência¹ se ocorrerem por volta dos 50ms, dos 50ms até 100ms e superiores a 100ms, após a apresentação do estímulo, respetivamente.

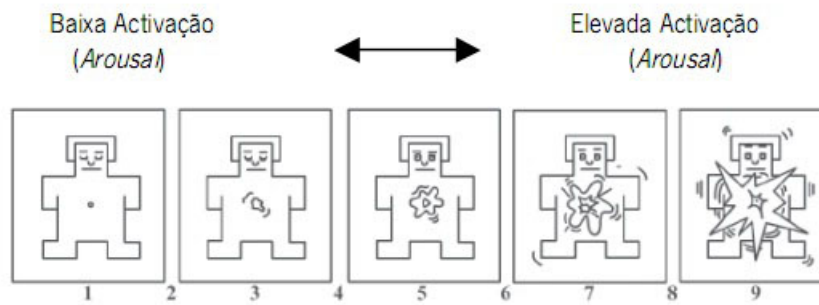
Os sinais de EEG utilizados neste projeto foram cedidos pela Psicóloga Mestre Ana Teresa Pereira, adquiridos para realização da sua tese de Mestrado [1]. Estes sinais apenas englobam estímulos visuais que foram selecionados da base de imagens IAPS [12]. Esta base contém 956 imagens classificadas em três dimensões emocionais: valência, arousal e dominância, avaliadas

¹é o tempo que separa o estímulo do aparecimento de um potencial característico

com recurso ao Self-Assessment Manikin (SAM). A valência é avaliada num contínuo que varia entre valência positiva e negativa, numa escala de 1 a 9, sendo o 1 correspondente à figura triste e significa que a imagem é muito desagradável e o 9 correspondente à figura alegre, isto é, uma imagem muito agradável (figura 2.1a). Da mesma maneira, a arousal também é medida numa escala que varia de 1, que corresponde a uma imagem que transmite calma ou relaxamento, a 9, que corresponde a uma imagem intensa o que deixa o sujeito num estado mais ativo ou excitado (figura 2.1b). A dominância não foi utilizada como critério para aquisição destes dados e as imagens utilizadas foram divididas em dois grupos: Grupo 1 - Elevado arousal e valência positiva e Grupo 2 - Elevado arousal e valência negativa.



(a)



(b)

Figura 2.1: Dispositivos utilizados para avaliação da valência - 2.1a - e da arousal - 2.1b dos estímulos visuais - imagem adaptada de [1]

2.3 Aquisição do sinal EEG

O sinal de EEG é adquirido através de elétrodos colocados no couro cabeludo. O sistema utilizado para colocação dos elétrodos para o registo do EEG foi o sistema internacional 10-20 (figura 2.2) ao qual foram adicionadas as derivações FPZ e OZ na linha central entre FP1 e FP2 e entre O1 e O2, relativamente na devida ordem. Os pontos pretos representam os elétrodos e as letras F, T, C, P e O são usadas para Frontais, Temporais, Centrais, Parietais e Occipitais, relativamente na devida ordem. A letra A funciona como referência aquando a aquisição do potencial. No que respeita aos números, os pares (2,4,6 e 8) estão associados ao hemisfério direito, os ímpares (1,3,5, e 7) estão associados ao hemisfério esquerdo e Z está associado à linha central do córtex cerebral [9]. A referência colocou-se na ponta do nariz.

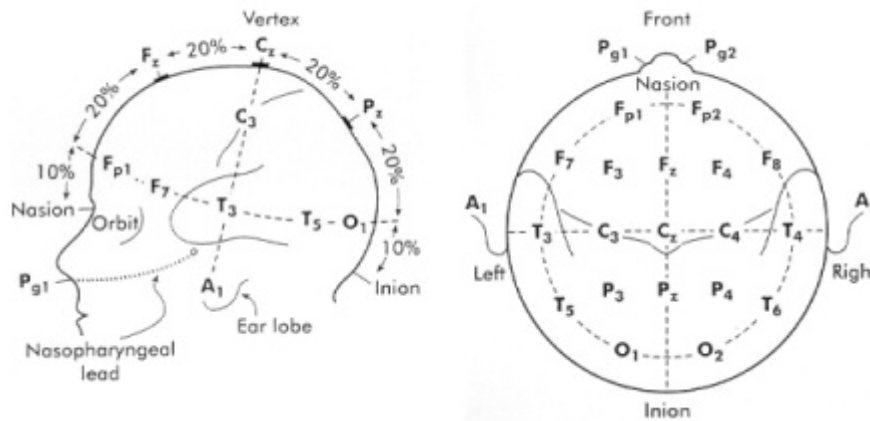


Figura 2.2: Representação da disposição de elétrodos segundo o sistema 10-20 (imagem adaptada de [5])

Para a aquisição dos sinais oculares VEOG e HEOG, colocaram-se os elétrodos de acordo o exposto na figura 2.3. Para o sinal ocular vertical (VEOG) foram colocados os elétrodos E1, E2, E3 e E4 que vão registar os pestanejos. Para o sinal ocular horizontal (HEOG), colocaram-se os elétrodos E5 e E6 para registar os movimentos horizontais dos olhos. A impedância foi mantida abaixo de $5K\Omega$ e a frequência de amostragem a 1000Hz, ou seja, 1000 amostras por segundo.

Na recolha de dados de EEG, o indivíduo esteve sentado a 80cm do ecrã do computador. Para evitar que pestanejasse durante a apresentação do estímulo, antes de cada imagem aparecia no ecrã um ponto de fixação (uma cruz branca num fundo preto) durante 750 ms (figura 2.4). A imagem era apresentada imediatamente depois durante 500 ms. O intervalo inter-ensaios foi de 3500 ms.

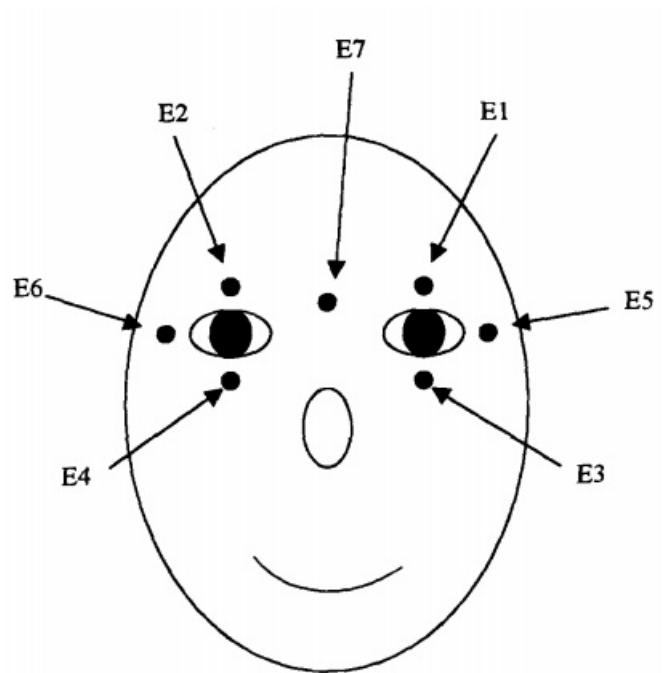


Figura 2.3: Posições dos eletrodos para aquisição dos sinais oculares (imagem adaptada de [4])

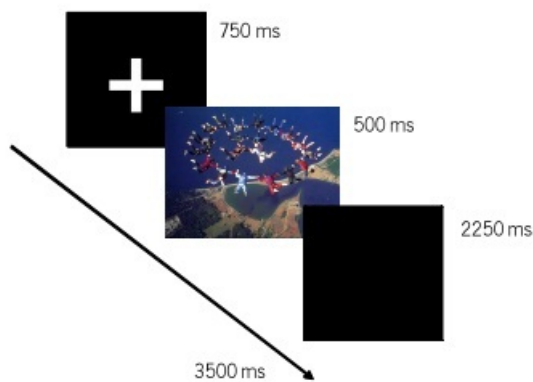


Figura 2.4: Esquema de apresentação dos estímulos no registo das medidas - imagem adaptada de [1]

2.4 Processamento de Potenciais Evocados

O sinal é registado continuamente num ficheiro, seguindo-se várias fases para obter diversos ficheiros necessários para análise (figura 2.5). A filtragem é a primeira fase por que passa o sinal contínuo, sendo utilizado um filtro passa banda de 0.1 Hz a 30Hz. Segue-se a eliminação de artefactos que não correspondam a artefactos oculares. Aqui são eliminados segmentos de sinal onde o artefacto se encontra. A correção dos movimentos oculares é a fase seguinte. Aqui, o programa corre canal a canal e sempre que considera artefacto pergunta ao utilizador

se é verdade. No final apresenta um quadro com os desvios-padrão de cada canal e a correção, que pode ser aceite ou não. Finalmente, o sinal é dividido em épocas² e para cada uma destas épocas é corrigida a linha de base e é calculada a sua média.



Figura 2.5: Vários passos por que passa o ficheiro contínuo registado inicialmente

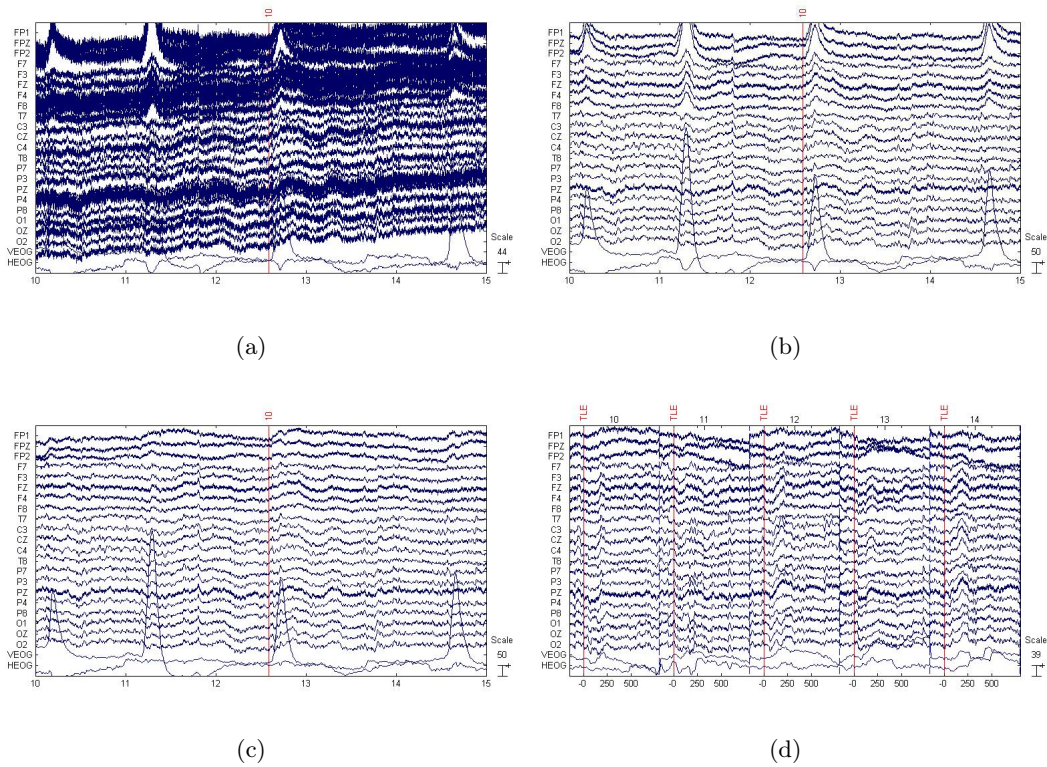


Figura 2.6: Passos para tratamento do sinal EEG: (a) apresenta o sinal contínuo sem qualquer tipo de tratamento; (b) apresenta o sinal depois de filtrado (0.1Hz - 30Hz); (c) apresenta o sinal depois de corrigido artefacto ocular; (d) apresenta o sinal depois de dividido em épocas e de corrigida linha de base

²Para a divisão em épocas destes dados foram considerados 150ms antes do estímulo e 800ms depois do estímulo.

2.5 Regressão linear na correção de artefatos

Os movimentos oculares (EOG) são a maior fonte de contaminação de um sinal de EEG. Estes dão origem, frequentemente, a ondas lentas e a picos de amplitude elevada, quando comparada com a amplitude normal de um sinal de EEG. Este tipo de artefactos contamina essencialmente os canais frontais ³. Na figura 2.7, é notória a interferência do movimento ocular no canal Fz do EEG, uma vez que o sinal considerado verdadeiro, ou seja, limpo de artefactos, quase não apresenta amplitude enquanto o medido chega aos -0.5uV de amplitude, principalmente no intervalo de tempo em que a amplitude do sinal ocular VEOG atinge o valor mais alto [4].

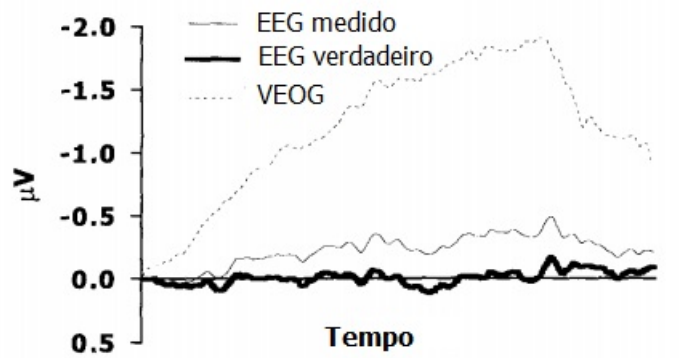


Figura 2.7: Efeito do movimento ocular no sinal de EEG, mais propriamente no canal Fz (imagem adaptada de [4])

A regressão linear é a técnica mais utilizada pelos métodos de correção de artefactos oculares. Este método baseia-se na subtração de parte do sinal de EOG vertical ($\alpha VEOG$) e parte do sinal de EOG horizontal ($\beta HEOG$) ao sinal de EEG medido: $EEG_{corrigido} = EEG_{medido} - \alpha VEOG - \beta HEOG$. Quando o EEG é capturado tendo como referência uma linha media central⁴, a influência do HEOG no sinal de EEG pode ser ignorado:

$$EEG_{corrigido} = EEG_{medido} - \alpha VEOG . \quad (2.1)$$

Uma vez que os dados utilizados foram adquiridas tendo como referência o nariz do indivíduo, vai-se ter em conta apenas o canal de VEOG

Tendo em conta que os registos do EEG são multicanal (digital), o canal m ($x_m(n)$) é corrigido da seguinte forma:

$$\hat{x}_m(n) = x_m(n) - ay(n) , \quad (2.2)$$

onde $x_m(n)$ e $\hat{x}_m(n)$ são os sinais de EEG contaminado e não contaminado pelo sinal VEOG, respetivamente; $y(n)$ é o sinal VEOG e a é a constante de proporção do sinal VEOG presente em cada canal do sinal EEG. Com isto é possível obter um sinal não contaminado pelo canal ocular, ou seja, limpo do artefacto ocular.

³Sensores situados na parte frontal do cerebro, como se verá mais à frente

⁴O exemplo mais comum é o nariz

Como anteriormente mencionado, a é a constante de proporção do sinal VEOG presente em cada canal do sinal EEG o que permite simplificar a equação 2.2 chegando a:

$$x_m(n) = ay(n) . \quad (2.3)$$

Para conseguir calcular a constante de proporção a o método regressivo recorre às regras da regressão linear 2.4 que são conseguidas de forma a minimizar o erro quadrático médio [13].

$$a = \frac{COV(y, x)}{Var[y]} = \rho_{yx} \frac{\sqrt{Var[x]}}{\sqrt{Var[y]}} , \quad (2.4)$$

onde COV é a covariância, Var é a variancia e $\rho_{yx} = \frac{COV(y, x)}{\sqrt{Var[x]}\sqrt{Var[y]}}$ é o coeficiente de correlação.

Este método é utilizado pelo Neuroscan⁵ mas combinado com a média dos artefactos. Isto é, o Neuroscan calcula um sinal y que é uma média de vários artefactos oculares. Com isto é calculada uma constante de propagação a para cada canal. No método regressivo, implementado neste trabalho, a constante de propagação é calculada para cada pico detetado e considerado movimento ocular, ou seja, não é feita uma média dos vários máximos encontrados, nem é usado o maior máximo para determinar os restantes movimentos oculares.

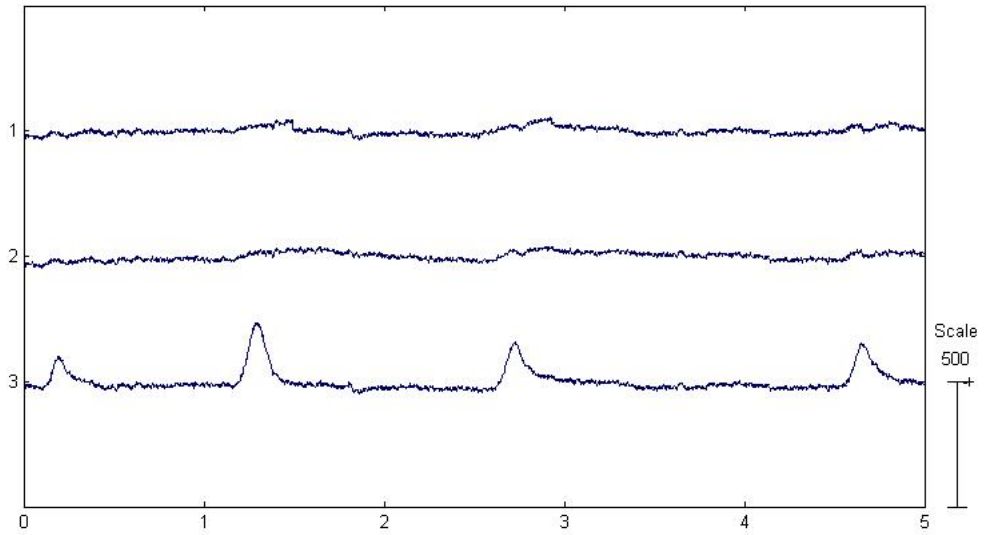


Figura 2.8: Canal Fz contaminado (3), corrigido pelo método regressivo (1) e corrigido pelo Neuroscan (2)

⁵é um programa de aquisição e tratamento do sinal de EEG mais utilizado e foi utilizado para aquisição dos sinais de EEG utilizados

2.6 Outros métodos de correção de artefactos oculares

Como alternativa à regressão linear simples, têm surgido outros métodos de correção de artefactos oculares. O método ICA e o método baseado em filtro adaptativo são dois exemplos desses métodos.

O ICA é um método de correção de artefactos oculares implementado no EEGLab. Este método utiliza dois algoritmos diferentes para remoção de artefactos oculares, o *Informax-ICA* [10] e o *Extended-Informax-ICA* [18]. A diferença entre eles é que este último consegue remover os artefactos oculares e o ruído provocado pelos 50Hz, enquanto o primeiro apenas remove os artefactos oculares. O principal objetivo do ICA passa por divisão do sinal em M componentes independentes de forma a isolar o artefacto ocular. Este método preserva as características não-lineares do sinal de EEG depois de remoção dos artefactos.

O método baseado em filtro adaptativo [14] usa o VEOG e o HEOG como duas entradas de referência. Cada entrada de referência é processada por um filtro com resposta impulsional finita de tamanho M e, de seguida, é subtraída ao sinal de EEG original. O método utiliza um algoritmo recursivo baseado em mínimos quadrados que inclui um fator de esquecimento (λ) para controlar a porção não-estacionária do sinal EOG.

2.7 Conclusões

Este capítulo abordou, de forma resumida, o que é um sinal de EEG e os artefactos/potenciais evocados que podem surgir neste tipo de sinais. Foi também abordada a técnica da regressão linear e a forma como ela é implementada quer no Neuroscan, quer nas técnicas aqui desenvolvidas. Ficou-se a perceber que o método de correção de artefactos oculares do Neuroscan utiliza um artefacto médio para assegurar que apenas retira o artefacto ocular do sinal EEG, e não informação relevante deste. No entanto, isto pode ser visto como uma desvantagem pois, como está a ter em conta uma média, pode não retirar, na sua totalidade, o artefacto ocular presente no sinal de EEG.

Um problema que surge com o uso do método regressivo é que este, como mencionado anteriormente, apenas admite que o sinal de EEG seja contaminado pelo sinal de EOG, contudo, está comprovado que o contrário também se verifica, ou seja, o sinal de EOG também é contaminado pelo sinal de EEG. Desta forma, ao subtrair ao sinal de EEG o sinal de EOG, pode-se estar a remover parte de sinal de EEG também.

O capítulo seguinte descreve outros dois métodos baseados neste método regressivo combinado com a divisão do sinal em componentes.

Capítulo 3

SVD e remoção de artefatos

3.1 Introdução

Um vasto número de métodos tem sido proposto para resolver a problemática da distorção que aparece nos sinais EEG devido a movimentos oculares ou pestanejos ocorridos na altura da aquisição do sinal. Estes métodos tornaram-se necessários uma vez que, tal distorção induz uma má interpretação dos sinais de EEG.

Anteriormente, foi mencionada a regressão linear como uma solução à correção de artefactos, neste capítulo irá falar-se de métodos que também utilizam a regressão linear, mas aplicada às componentes, ou seja, métodos que transformam os sinais em componentes e compara estas com o canal ocular VEOG. Isto consegue-se recorrendo à decomposição em valores singulares (SVD) [3].

3.1.1 Análise multicanal - PCA

Análise de componentes principais (PCA) é um dos principais pilares da análise de dados moderna. A análise PCA é definida matematicamente como uma transformação ortogonal linear que transforma dados correlacionados em novas coordenadas do sistema não correlacionadas, as componentes principais. Estas componentes são individualmente responsáveis pela variância dos dados. A vantagem de tornar estes dados não correlacionados é que elimina parte da informação redundante em cada dimensão, o que serve para realçar as semelhanças e diferenças dos dados, ou seja, os padrões existentes.

Um segmento de sinais EEG representa-se por uma matriz \mathbf{X} de dimensões $[M \times N]$, ou seja, M canais e N amostras que pode ser decomposta nas suas principais componentes. A matriz de correlação $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ apresenta como suas entradas as correlações entre os canais de EEG (as linhas da matriz \mathbf{X} - figura 3.1). Através da decomposição em valores próprios da matriz \mathbf{S} obtêm-se:

$$\mathbf{S} = \mathbf{U}\mathbf{D}\mathbf{U}^T, \quad (3.1)$$

onde \mathbf{U} é a matriz ortogonal ($\mathbf{U}^T\mathbf{U} = \mathbf{I}$ e $\mathbf{U}\mathbf{U}^T = \mathbf{I}$) que contém os vetores próprios e $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$ é uma matriz diagonal que contém os valores próprios. A matriz \mathbf{U} vai permitir a transformação dos dados correlacionados da matriz inicial \mathbf{X} em dados não correlacionados: $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$. Cria-se, assim, uma nova matriz \mathbf{Z} (figura 3.2b).

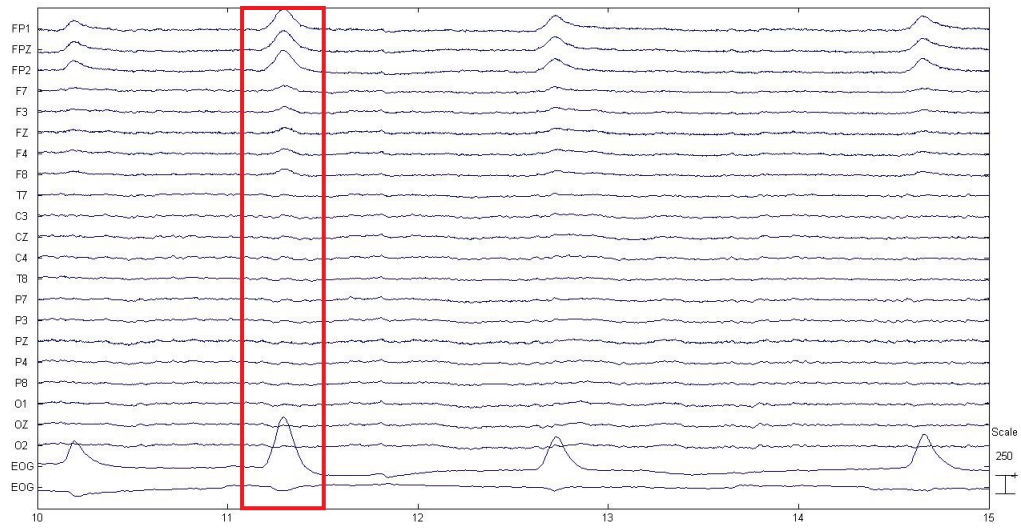


Figura 3.1: Sinal de EEG utilizado (10 - 15 segundos)

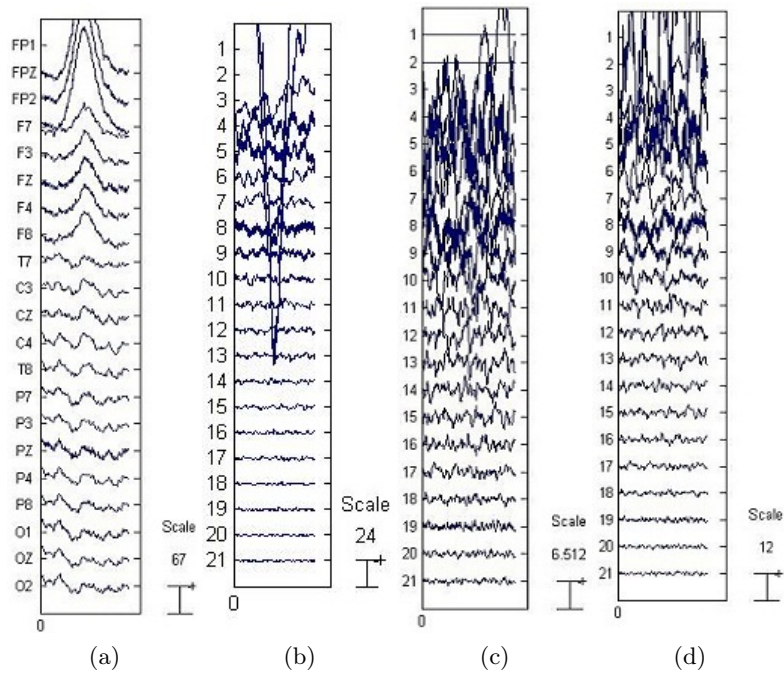


Figura 3.2: Método multicanal aplicado a um pestanejo (400ms): (a) - Pestanejo marcado na figura 3.1; (b) - Pestanejo depois de transformado em componentes principais; (c) - Rejeição de artefacto aplicado às duas primeiras componentes; (d) - Correção de artefacto aplicado às duas primeiras componentes

Como mencionado anteriormente, esta transformação permite realçar as semelhanças e diferenças nos dados. Na figura 3.2b é realçada a semelhança das primeiras componentes com

o sinal ocular. Para verificar as semelhanças entre cada componente e o sinal ocular recorre-se ao coeficiente de correlação e a um limiar definido inicialmente.

A cada componente que seja considerada semelhante ao canal ocular aplica-se a regressão linear (figura 3.2d) ou, simplesmente, elimina-se a componente (figura 3.2c).

Finalmente, depois de corrigida/rejeitada cada componente é necessário reconstruir o sinal e, para isso, recorre-se à equação 3.2. Obtém-se, assim, o sinal inicial limpo do artefacto ocular detetado (figura 3.3)

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{Z} . \quad (3.2)$$

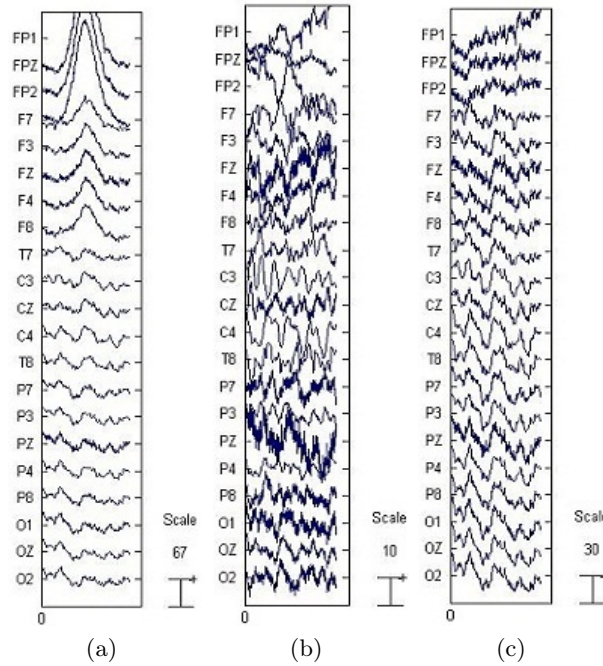


Figura 3.3: (a) - Pestanejo marcado na figura 3.1 ; (b) - Pestanejo depois de rejeitado artefacto ocular; (c) - Pestanejo depois de corrigido artefacto ocular

Existem, no entanto algumas desvantagens associadas a este método, tais como a exigência de que os artefactos sejam não correlacionados com o sinal de EEG [6]. Tem sido demonstrado que PCA não consegue separar completamente os artefactos oculares do sinal de EEG, principalmente quando têm amplitudes semelhantes. Para além disto, as características semelhantes do sinal de EEG não são necessariamente decompostas dentro das mesmas componentes quando aplicadas a diferentes épocas.

3.1.2 Análise canal a canal - SSA

A análise espectral simples (SSA) [16] é uma poderosa técnica de análise de séries temporais incorporando os elementos da análise clássica de séries temporais, estatísticas multi-variadas, geometria multi-variada, sistemas dinâmicos e processamento de sinal. O objetivo desta análise é decompor as séries temporais originais na soma de componentes não-correlacionadas,

tais como uma tendência de variação lenta, componentes oscilatórios e ruído.

Tal como o método PCA, o SSA recorre à decomposição em valores singulares (SVD), porém, apenas trata canal a canal e não todos os canais como acontece no método PCA. Considerando um segmento de um canal EEG ($x[0], x[1], x[2], \dots, x[N-1]$) (figura 3.4a), um sinal multidimensional é obtido através de $\mathbf{x}_k = (x[k-1+M-1], \dots, x[k-1])^T, k = 1, \dots, K$. Desta forma, no método SSA a matriz \mathbf{X} é dada por:

$$\mathbf{X} = \begin{bmatrix} x[M-1] & x[M] & \dots & x[N-1] \\ x[M-2] & x[M-1] & \dots & x[N-2] \\ x[M-3] & x[M-2] & \dots & x[N-3] \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x[0] & x[1] & \dots & x[N-M] \end{bmatrix} \quad (3.3)$$

Esta matriz apresenta uma estrutura ‘Toeplitz’¹. Este método recorre, assim, a um procedimento de atraso para transformar uma série temporal uni-dimensional numa matriz multi-dimensional. Na figura 3.4 está presente o pestanejo utilizado para correção e três diferentes valores para o M (número de atrasos), escolhidos aleatoriamente.

Em seguida, como no método PCA, constrói-se a matriz de correlação (\mathbf{S}) com a mesma finalidade mencionada anteriormente. Porém, aqui as entradas da matriz \mathbf{S} representam a correlação entre amostras e não entre canais como acontecia no método anterior. Calculando os M vetores próprios $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_M]$, pode-se escrever a matriz \mathbf{X} como uma soma de matrizes:

$$\mathbf{X} = \mathbf{u}_1 \mathbf{u}_1^T \mathbf{X} + \mathbf{u}_2 \mathbf{u}_2^T \mathbf{X} + \dots + \mathbf{u}_M \mathbf{u}_M^T \mathbf{X} = \hat{\mathbf{X}}_1 + \hat{\mathbf{X}}_2 + \dots + \hat{\mathbf{X}}_M . \quad (3.4)$$

Nas matrizes ($\hat{\mathbf{X}}_m, m = 1, \dots, M$) os elementos de cada diagonal já não são iguais como acontecia na matriz \mathbf{X} . Uma matriz de Toeplitz pode ser obtida para as matrizes $\hat{\mathbf{X}}_m$, substituindo cada elemento na diagonal pela sua média. De cada matriz Toeplitz, um segmento de N amostras com os valores da primeira coluna e da primeira linha da matriz (ver equação 3.3) constitui uma componente $\hat{x}_m[n]$ do segmento original $x[n]$.

O sinal original pode ser reconstruído com a soma destas componentes

$$\hat{x}[n] = \sum_m \hat{x}_m[n] .$$

Para remover artefactos no sinal, as componentes $\hat{x}_m[n]$ relacionadas com o pestanejo são eliminadas (figura 3.5) ou é-lhes aplicada a regressão linear (figura 3.6). Desta forma, o sinal $\hat{x}[n]$ é o sinal com o artefacto corrigido (figura 3.7 e 3.8).

¹Matriz que contém os elementos de cada diagonal iguais.

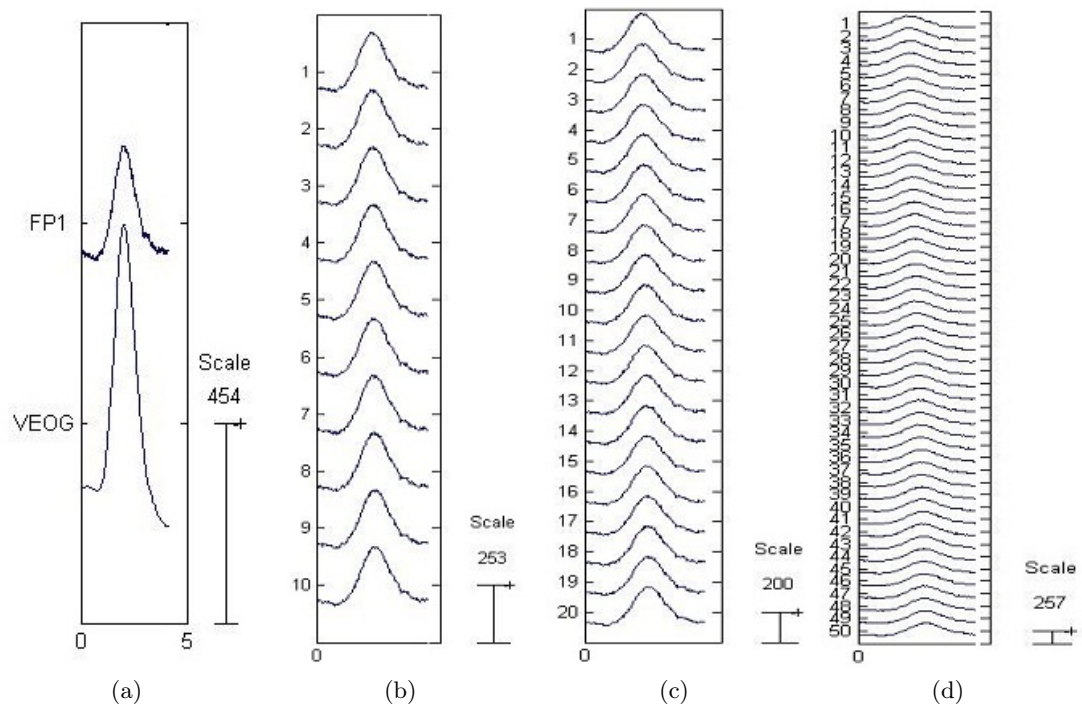


Figura 3.4: Canal a tratar e respectivos atrasos temporais (400ms): (a) - Pestanejo marcado na figura 3.1 (Canal FP1); (b) - $M=10$ - Sinal depois de atrasado 10 vezes (convertido em 10 componentes); (c) - $M=20$ - Sinal depois de atrasado 20 vezes (convertido em 20 componentes); (d) - $M=50$ - Sinal depois de atrasado 50 vezes (convertido em 50 componentes)

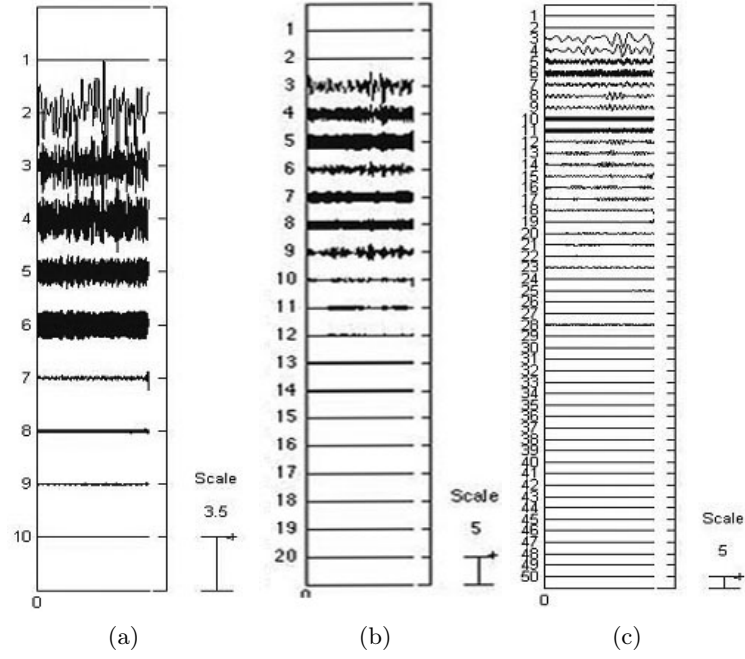


Figura 3.5: Aplicação do método com rejeição do artefacto: (a) $M=10$; (b) $M=20$; (c) $M=50$

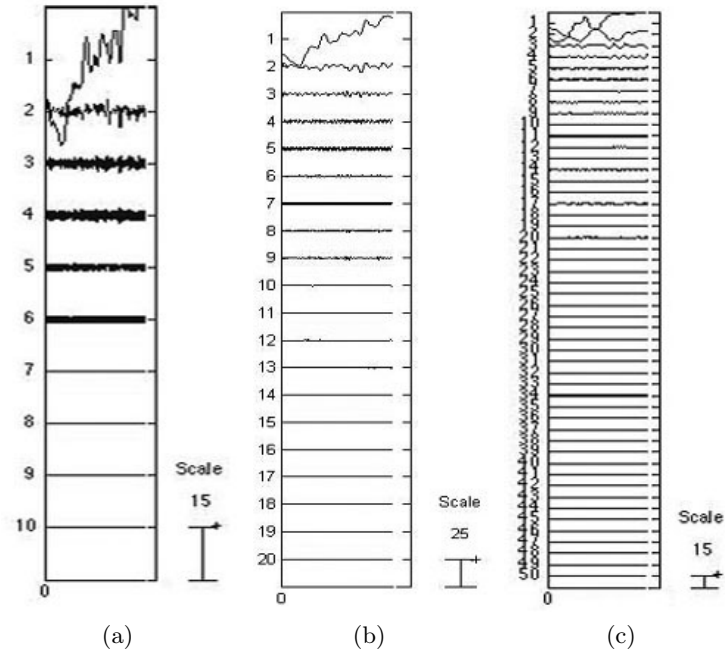


Figura 3.6: Aplicação do método com correção do artefacto à matriz \mathbf{X} : (a) $M=10$; (b) $M=20$; (c) $M=50$.

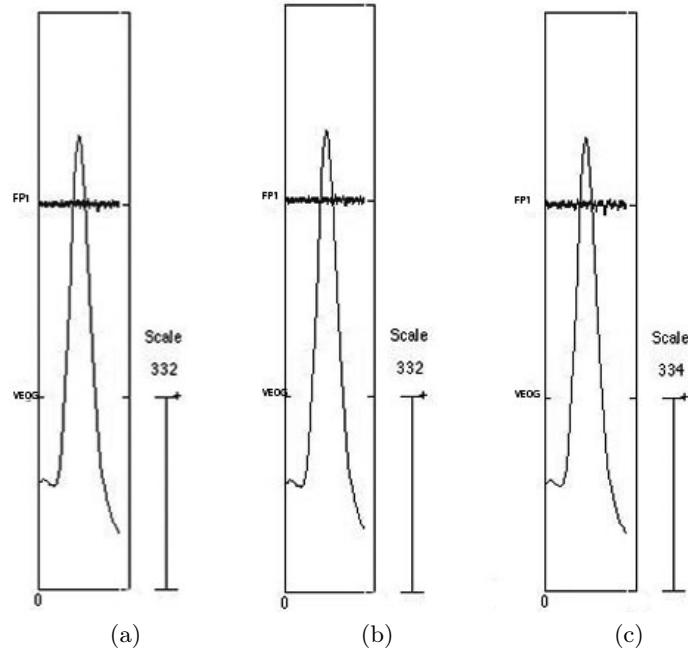


Figura 3.7: Sinal recuperado depois de aplicação do método com opção de rejeição do artefacto: (a) $M=10$; (b) $M=20$; (c) $M=50$.

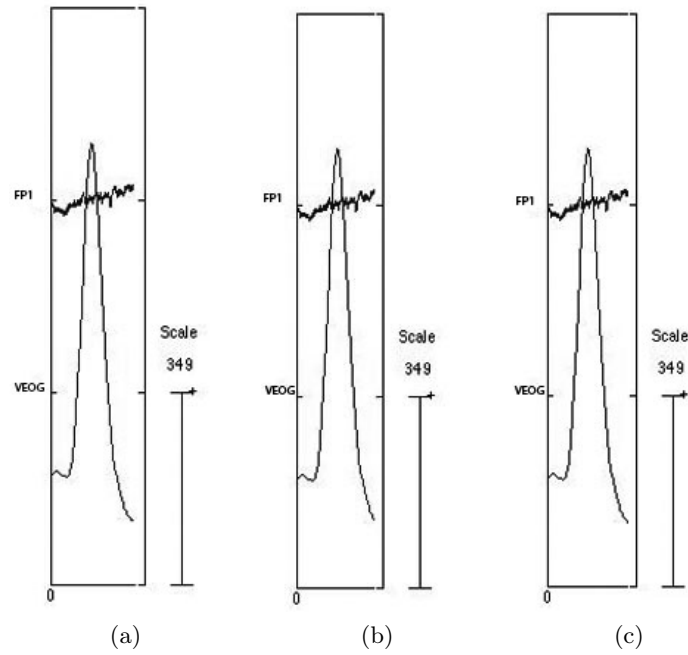


Figura 3.8: Sinal recuperado depois de aplicação do método com opção de correcção do artefacto: (a) $M=10$; (b) $M=20$; (c) $M=50$.

3.2 Conclusões

Neste capítulo foram descritos dois métodos baseados em SVD aplicados em segmentos de sinal onde há artefactos. O primeiro método utiliza todos os canais de EEG para corrigir um segmento de sinal com artefacto, enquanto que o segundo é aplicado em paralelo a cada canal. No segundo método, o número de componentes é atribuído de acordo com o que produz uma melhor correção do artefacto, o que pode incutir erros nos resultados se a escolha for mal feita. Ambos os métodos apresentam as suas primeiras componentes de acordo com os seus valores próprios mais elevados e normalmente são as mais distorcidas pelo artefacto ocular.

O capítulo seguinte descreve um sistema que pretende corrigir automaticamente os sub-segmentos, afetados por artefactos, de um sinal contínuo. Mas para isso será necessário, de uma forma automatizada, detetar os artefactos, o que vai permitir corrigir apenas o sub-segmento do sinal pretendido.

Capítulo 4

O projecto do sistema automático

4.1 Introdução

Este capítulo é dedicado especificamente à criação de um sistema automático para correção de movimentos oculares em sinais de EEG. Na primeira secção (“Descrição global do sistema”) são apresentados os dois módulos em que se divide o sistema desenvolvido. Nas duas secções seguintes (“Interface gráfica” e “Módulo de processamento”) é feita uma descrição mais pormenorizada de cada módulo do sistema, nomeadamente a escolha de parâmetros e a explicação das funções usadas para criação do software. O capítulo termina com a secção “Conclusões”, onde é feita a avaliação do sistema desenvolvido mencionando as vantagens que surgem com os módulos utilizados.

4.2 Descrição global do sistema

O sistema automático desenvolvido encontra-se dividido em dois módulos (ver figura 4.1). Estes dois módulos podem ser executados de forma independente sendo, por isso, necessário proceder a uma descrição mais detalhada de cada um deles.



Figura 4.1: Diagrama de blocos geral do sistema

4.3 Interface gráfica

A interface gráfica presente no sistema é uma interface simples que apenas foi construída através da utilização das funcionalidades GUI do Matlab para uma melhor avaliação dos métodos implementados e do módulo de processamento em geral. A interface é dividida em vários sub-módulos, que permitem ao utilizador escolher os vários parâmetros necessários para o

módulo de processamento.

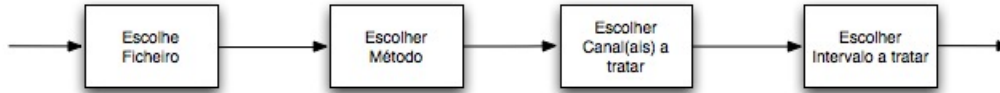


Figura 4.2: Diagrama de blocos da interface gráfica desenvolvida

Numa primeira fase, é pedido ao utilizador o ficheiro que quer analisar dando a possibilidade de visualizar o sinal presente no ficheiro. Desta forma, permite ao utilizador conhecer o sinal antes de ser tratado e escolher o segmento do sinal a corrigir. Tendo em conta o objetivo desta interface gráfica mencionado anteriormente, o sistema apenas admite para tratamento sinais de EEG adquiridos em 21 canais de EEG e 2 canais de EOG em formato contínuo e, de preferência, já filtrados.

A fase seguinte consiste em escolher o intervalo de tempo do sinal EEG que se pretende analisar, assim como escolher o método a utilizar e, dependendo deste, os canais que se pretende corrigir e optar por corrigir ou eliminar componentes do sinal (ver figura 4.3).

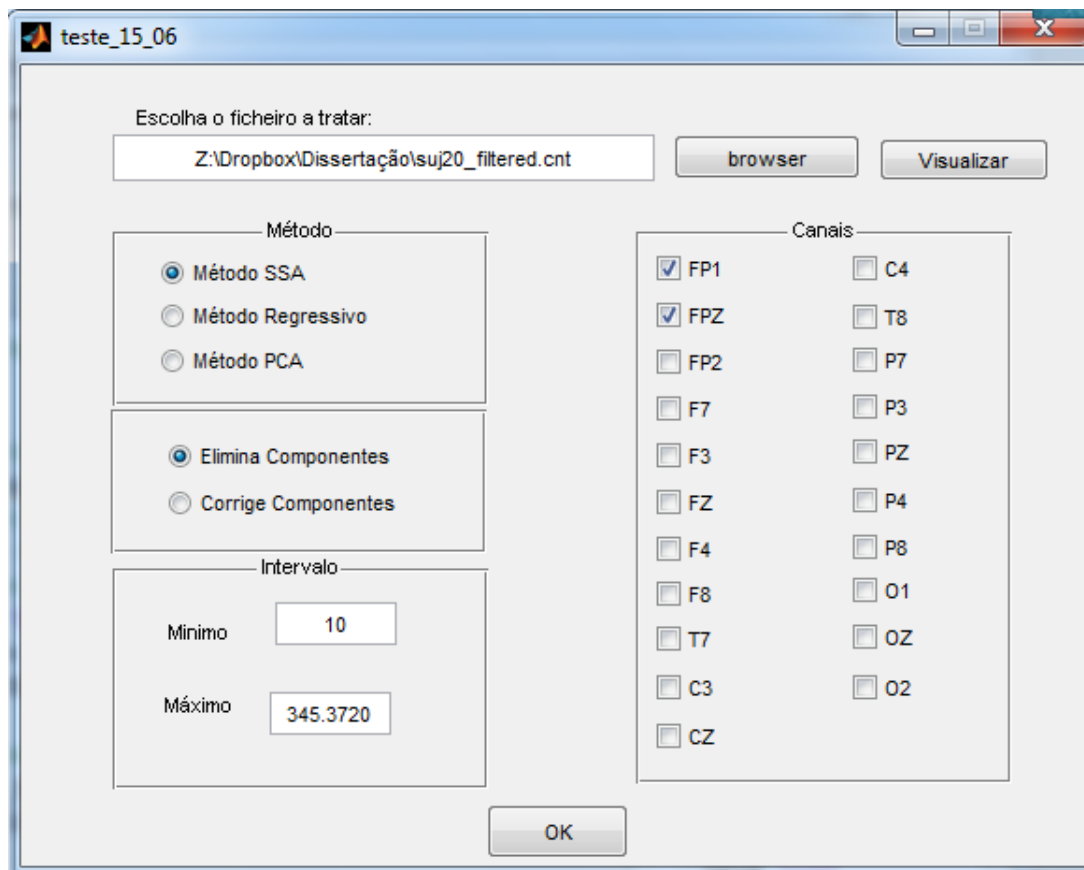


Figura 4.3: Interface gráfica desenvolvida

Depois de efetuada a correção pretendida dos artefactos oculares, a interface mostra ao

utilizador o sinal final. Aqui o utilizador vai conseguir visualizar se a correção está de acordo com o pretendido e, caso não esteja, pode voltar a escolher todos os parâmetros anteriormente descritos. Desta forma é possível testar a aplicação dos diferentes métodos e dos diferentes parâmetros de correção.

4.3.1 Software da interface desenvolvido

Todo o software para criar a interface resulta da utilização de funcionalidades GUI do Matlab e de funções da EEGLab¹. No apêndice B.1 encontra-se o código associado à função da interface.

- Funções Gui Matlab: Estas funções foram utilizadas para criar o aspeto gráfico da interface.
- Funções da toolbox EEGLab: Foram utilizadas funções como o `pop_loadcnt` que permitem carregar o ficheiro contínuo e decompô-lo em estruturas (mais adiante falar-se-á sobre isso) e o `eegplot` que é responsável pela visualização do sinal de EEG.

$$EEG = pop_loadcnt(file, 'dataformat', 'int32')$$

Parâmetros de entrada:

- `file`: Variável que contém o caminho para o ficheiro escolhido;
- `'dataformat'` e `'int32'`: indica a função que o ficheiro está no formato de 32 bits.

Parâmetros de saída:

- `EEG`: estrutura de dados criada pela EEGLab.(ver apêndice A)

$$eegplot(EEG.data, 'srate', EEG.srate, 'elocfile', EEG.chanlocs, 'events', EEG.event)$$

Parâmetros de entrada:

- `EEG.data`: classe da estrutura EEG que contém os dados do sinal;
- `'srate'` e `EEG.srate`: indica a função que a frequência de amostragem do sinal;
- `'eloc_file'` e `EEG.chanlocs`: indica à função a classe da estrutura EEG que contém o nome dos canais do sinal de EEG;
- `'events'` e `EEG.event`: indica à função a classe da estrutura EEG que contém os estímulos ocorridos no sinal de EEG.

4.4 Módulo de Processamento

O módulo de processamento do sistema divide-se em duas fases importantes: a deteção de movimentos oculares e a aplicação do método ao movimento ocular detetado (figura 4.4).

¹Toolbox do Matlab que auxiliar na análise e manuseamento de sinais EEG



Figura 4.4: Diagrama de blocos do módulo de processamento

4.4.1 Detetor de movimentos oculares

O detetor de movimentos oculares é simplesmente um detetor de picos do sinal ocular. É feito um varrimento ao canal ocular (VEOG) e guardado em dois registos o valor e o índice de cada mínimo e máximo detetado. Esta deteção é feita recorrendo à função *peakdet*² presente no Apêndice B.2.

$$[maxtab \ mintab] = peakdet(v, delta, x)$$

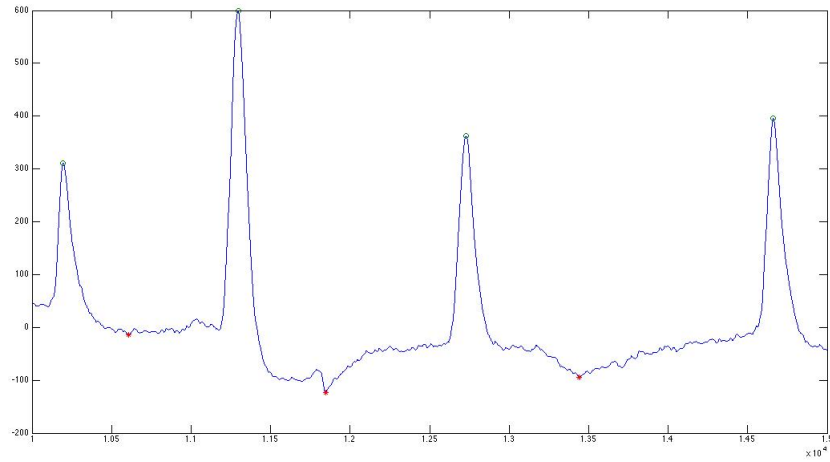
Parâmetros de Entrada:

- *v*: Sinal em que se pretendem detetar os máximos e mínimos;
- *delta*: Valor que vai definir se o valor encontrado é máximo/mínimo. Todo o ponto é máximo se tiver o máximo valor e se fora seguido por um valor menor que delta e todo o ponto é mínimo se tiver o mínimo valor e se fora seguido por um valor maior que delta;
- *x*: intervalo de tempo do sinal que se pretende fazer a deteção.

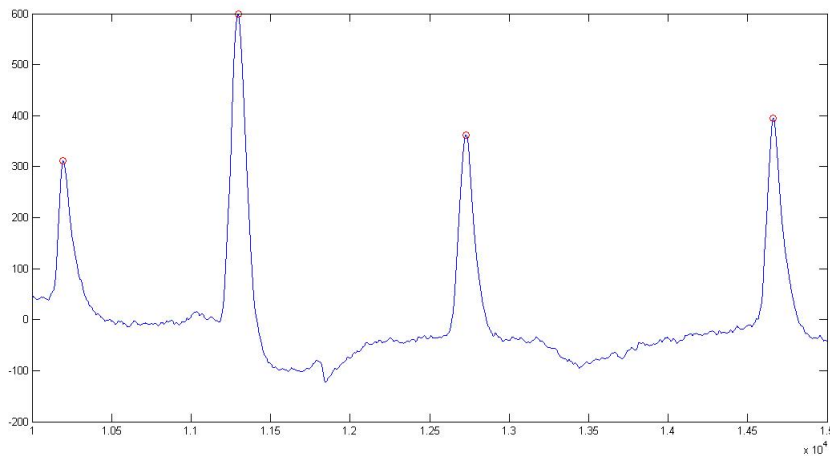
Parâmetros de Saída:

- *maxtab*: Matriz de duas colunas onde são guardados os índices e o valor dos máximos detetados;
- *mintab*: Matriz de duas colunas onde são guardados os índices e o valor dos mínimos detetados.

²função retirada de [2]



(a)



(b)

Figura 4.5: (a) Detector de picos (máximos e mínimos) ; (b) Movimentos oculares apenas com a detecção dos máximos.

Depois de detetados os movimentos oculares presentes no sinal ocular VEOG procede-se à análise da interferência desses movimentos nos canais de EEG. Para isso, tem-se em conta uma duração do pestanejo de 400ms, ou seja, com a tabela dos índices dos picos processam-se janelas de 400ms dos sinais EEG.

Note-se que nem todos os movimentos oculares vão gerar artefactos oculares e a sua interferência difere de canal para canal. Esta avaliação da existência de artefacto ocular é feita (como anteriormente descrito) recorrendo à correlação entre canais e a um limiar previamente definido.

4.4.2 Restante software desenvolvido

Todo o restante software para criar o módulo de processamento consiste na implementação dos métodos de correção utilizados neste sistema. Estes métodos são chamados através de uma função principal que é chamada pela interface desenvolvida (ver apêndice B.3).

Método Regressivo

Este método não é nada mais nada menos que uma aplicação das regras de regressão linear (como já anteriormente descrito). No apêndice B.4 encontra-se o código associado a este método.

$$Y = \text{metodoregressivo}(y, x);$$

Parâmetros de Entrada:

- y : Sinal de EEG contaminado (apenas 400ms - duração do pestanejo);
- x : Sinal ocular VEOG (apenas 400ms - duração do pestanejo);

Parâmetros de Saída:

- Y : Sinal de EEG depois de corrigido artefacto ocular (apenas 400ms - duração do pestanejo);

Método Análise Multicanal

O código do método da análise multicanal (PCA) pode ser encontrada no apêndice B.5 e baseia-se na seguinte função:

$$[X, dd] = \text{pcasimples}(Z, x, flag)$$

Parâmetros de Entrada:

- Z : Matriz $M \times N$ - M canais de EEG e N amostras - (apenas 400ms - duração do pestanejo)
- x : Sinal ocular VEOG (apenas 400ms - duração do pestanejo);
- $flag$: Variável que indica à função a opção de corrigir ou rejeitar o artefacto ocular.

Parâmetros de Saída:

- X : Sinal de EEG ($M \times N$) depois de corrigido artefacto ocular (apenas 400ms - duração do pestanejo)
- dd : Variável de controlo dos coeficientes de correlação.

Método Análise canal a canal

Por fim, o método de análise canal a canal (SSA) recorre a algumas funções explicadas de seguida e cujo código Matlab se pode encontrar no apêndice B.6.

$$[z, dd] = ssa(x, M, y, flag)$$

Parâmetros de Entrada:

- x : Matriz $1 \times N$ (N amostras) que representa o sinal de EEG - (apenas 400ms - duração do pestanejo)
- M : Numero de atrasos que o sinal sofre;
- y : Sinal ocular VEOG (apenas 400ms - duração do pestanejo);
- $flag$: Variável que indica à função a opção de corrigir ou rejeitar o artefacto ocular.

Parâmetros de Saída:

- X : Sinal de EEG ($1 \times N$) depois de corrigido artefacto ocular e de somadas as componentes (apenas 400ms - duração do pestanejo)
- dd : Variável de controlo dos coeficientes de correlação.

Dentro da função anterior existem outras funções como a *newdate* que é utilizada para fazer o atraso do sinal inicial, isto é converte um vetor $1 \times N$ para uma matriz $M \times N$ e e *joinO* que transforma uma matriz num sinal. O código destas funções pode ser visto no apêndice B.6 e consistem:

$$X = newdate(x, M)$$

Parâmetros de Entrada:

- x : Matriz $1 \times N$ (N amostras) que representa o sinal de EEG - (apenas 400ms - duração do pestanejo)
- M : Número de atrasos que o sinal sofre;

Parâmetros de Saída:

- X : Sinal de EEG depois de sofrer um atraso de M (matriz $M \times N$) (apenas 400ms - duração do pestanejo)

$$x = joinO(X)$$

Parâmetros de Entrada:

- X : Matriz $M \times N$ depois de convertida em componentes (apenas 400ms - duração do pestanejo)

Parâmetros de Saída:

- x : Sinal construído através da matriz X (apenas 400ms - duração do pestanejo)

4.5 Conclusões

Este capítulo descreveu a estrutura global do sistema automático desenvolvido para correção de movimentos oculares presentes em sinais de EEG. O sistema está organizado em dois módulos (interface gráfica e processamento).

Uma das maiores vantagens deste sistema é que os módulos podem ser executados independentes, ou seja, o módulo de processamento pode ser utilizado sem necessidade de recorrer à interface gráfica. Isto torna-se uma vantagem, pois permite ao utilizador, caso possua conhecimentos para isso, fazer uma análise do sinal de forma mais complexa, não se limitando às funcionalidades implementadas na interface. Contudo, apesar de a interface desenvolvida ter sido apenas para facilitar a avaliação dos métodos implementados como anteriormente mencionado, a verdade é que as funcionalidades GUI do Matlab podem ser muito úteis na criação de uma interface mais profissional e complexa.

Outra vantagem do sistema é o facto de recorrer a funções da toolbox EEGLab. Esta toolbox, para além de apresentar funções que permitem trabalhar de uma forma gráfica, também dá a opção de serem utilizadas implícitas noutras funções, o que para este sistema foi necessário e útil. Para além disto, como mencionado anteriormente, utiliza estruturas de dados para classificar e agrupar os parâmetros do sinal carregado. Isto torna-se vantajoso, pois permite que o sistema seja feito de uma forma mais generalizada e para sinais com características diferentes. Existe ainda uma última vantagem que diz respeito à forma implementada do detetor de movimentos oculares. Este deteta tanto máximos como mínimos presentes no sinal, o que se pode tornar vantajoso pois existem vários sinais de EEG em que o sinal de VEOG aparece invertido devido à inversão da polaridade dos sensores que fazem esta medição. Assim, os movimentos oculares aí presentes vão ser detetados como mínimos e não como máximos. Daí a importância da possibilidade se detetar máximos e mínimos no sinal.

No capítulo seguinte, vai ser possível avaliar os diferentes métodos de correção de artefactos oculares utilizados neste sistema e compará-los entre si. Vai ser possível também comprovar todas as afirmações que foram feitas ao longo dos capítulos, através de resultados obtidos da realização de vários testes ao sistema desenvolvido.

Capítulo 5

Resultados

5.1 Introdução

Para uma avaliação das técnicas de correção de artefactos oculares, foram retirados vários resultados. Para isso recorreu-se à análise de dois registos diferentes de sinais EEG. O primeiro registo apresenta contaminação ocular apenas nos canais frontais, enquanto que no segundo registo há contaminação em todos os canais. É efetuada uma análise tempo-frequência (espectrograma) com a finalidade de visualmente ilustrar o impacto da técnicas de correção nos sinais. O impacto da correção nos sinais é avaliado calculando o coeficiente de correlação (entre os pares de sinais em estudo).

5.2 Aplicação das técnicas de correção

O processo de correção dos artefactos segue um padrão inicial igual para qualquer das técnicas de correção aplicadas. Inicialmente é feita uma deteção dos pestanejos presentes no sinal ocular VEOG. As diferentes técnicas para correção são depois aplicadas apenas a 400ms centrados no valor máximo de pestanejo detetado. Estes trechos vão ser “colados” ao sinal original.

Resumindo, o sinal de EEG pode ser corrigido com as seguintes técnicas:

- Regressão linear aplicada diretamente ao sinal original (MR)
- Método PCA com eliminação de componentes com valor de correlação superior a 0.5 (PCA - EL)
- Método PCA com correção das componentes correlacionadas com VEOG utilizando a regressão linear (PCA - CORR);
- SSA com eliminação de componentes com valor de correlação superior a 0.5 (SSA - EL);
- SSA com correção das componentes correlacionadas com VEOG utilizando a regressão linear regressão linear (SSA - CORR).

5.3 Características do Registo 1

Os sinais aqui analisados são sinais que seguem o protocolo descrito no capítulo “Artefactos e EEG” e são denominados como sendo simples uma vez que o registo foi efetuado de acordo com o protocolo experimental descrito, torna-o mais fácil de corrigir os artefactos oculares presentes. Apresentam uma duração de 345 segundos mas apenas será analisado o segmento dos 10 - 338 segundos devido a ruído inserido no início e no fim do sinal. Foram adquiridos em 23 canais (21 canais de EEG e 2 canais oculares) e contem 71 estímulos. Ao ser feita a deteção de movimentos oculares no intervalo de tempo acima mencionado, foram detetados 171 movimentos oculares (ver figura 5.1). Estes movimentos vão interferir com o sinal EEG, como é visível na figura 5.2

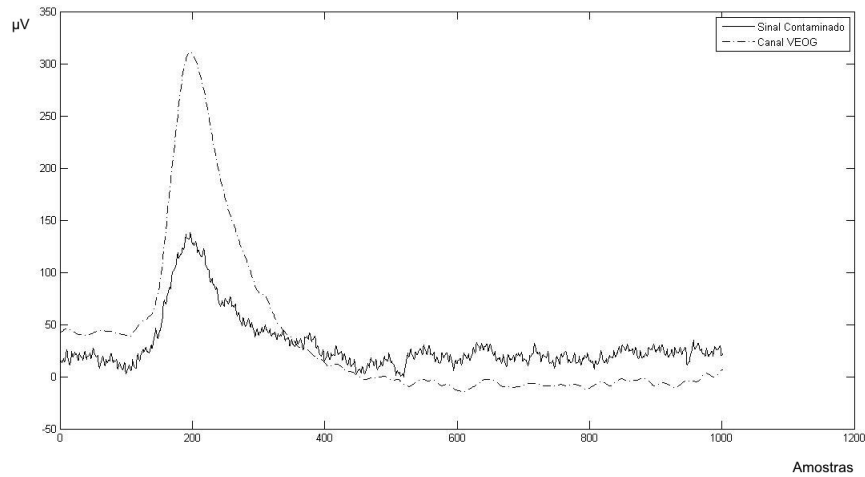


Figura 5.1: Canal FP1 e Canal VEOG (10 -11 segundos)

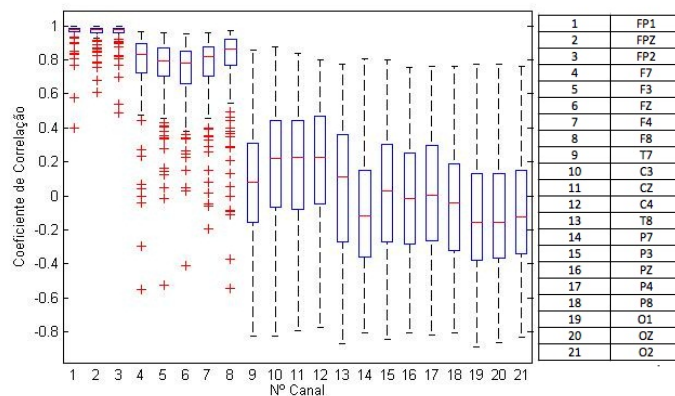


Figura 5.2: Caixa de bigodes dos coeficientes de correlação entre o sinal contaminado e o canal ocular VEOG calculados nos 400ms de cada pestanejo

5.3.1 Análise no tempo das técnicas implementadas

Quando o método regressivo é aplicado ao sinal contaminado, os artefactos oculares são removidos, o que significa que o sinal deixa de ter semelhanças com o sinal ocular (VEOG) o que leva a uma correlação praticamente nula entre estes dois sinais (ver figura 5.3a). No que respeita à correlação entre o sinal corrigido e o sinal contaminado, pode-se verificar que a correlação é menor nos canais frontais (até ao canal 8). Isto acontece pois estes canais eram os mais afetados pelo artefacto ocular. Os restantes canais apresentam coeficientes de correlação perto da unidade, o que significa que praticamente não sofreram alterações.

Na figura 5.4 pode-se comparar o sinal ocular VEOG com o sinal antes e depois de corrigido com o MR.

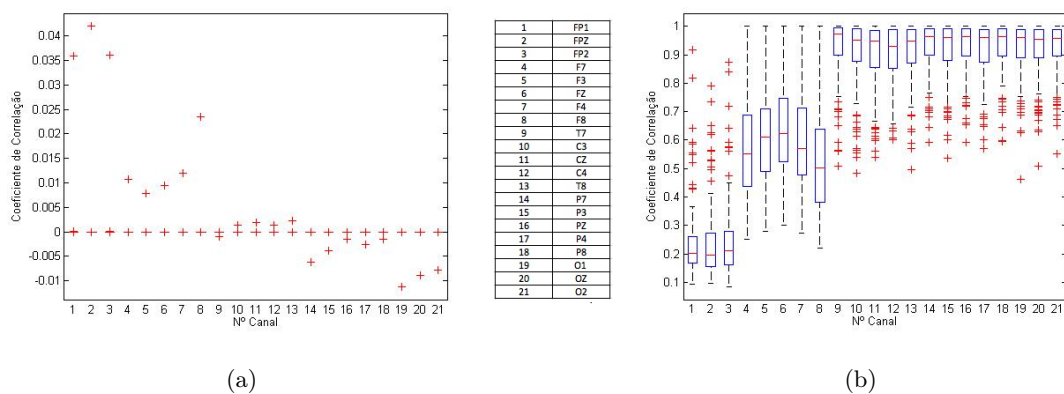


Figura 5.3: Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo MR: (a) com o canal ocular VEOG e (b) com o sinal contaminado, calculados nos 400ms de cada pestanejo.

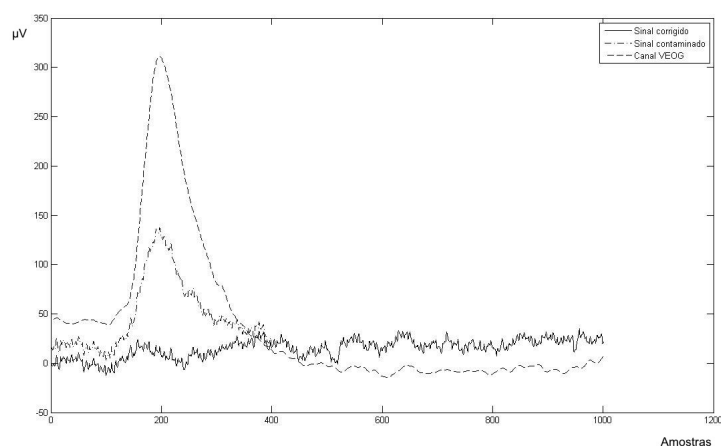


Figura 5.4: Comparação de sinais com canal FP1 depois de aplicado MR(10 -11 segundos)

Assim como acontece anteriormente, também para o método PCA - CORR foram obtidos resultados semelhantes como se pode ver nas figuras 5.5 e 5.6. Com este método os coeficientes

de correlação entre o sinal corrigido e o canal ocular VEOG mantêm-se praticamente nulos, contudo, com um ligeiro aumento relativamente ao Método Regressivo. Já a correlação com o contaminado, os resultados são semelhantes com os obtidos para o Método Regressivo. Com o método PCA - EL (5.5c e 5.5d) verifica-se uma maior variação dos coeficientes de correlação com o sinal ocular e diminui o valor dos coeficientes de correlação com o sinal contaminado. Isto porque o sinal perde informação com a eliminação total da componente.

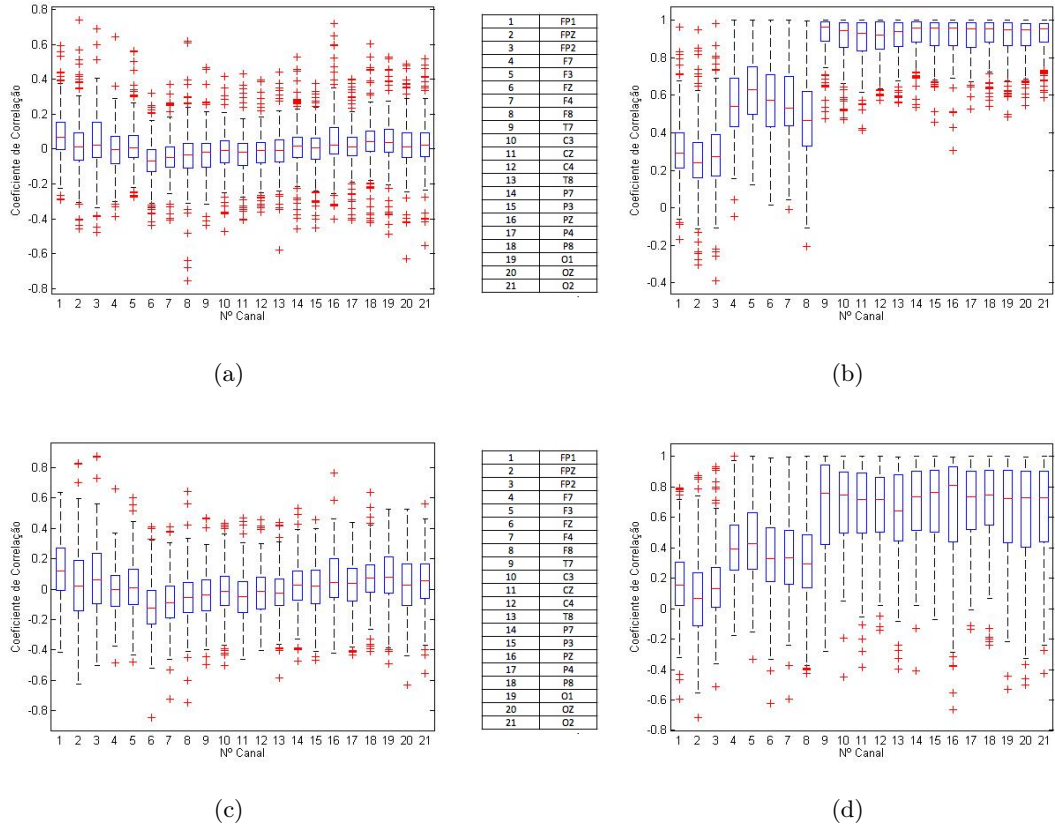


Figura 5.5: Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método: PCA - CORR: (a) com o canal ocular VEOG e (b) com o sinal contaminado; PCA - EL: (c) com o canal ocular VEOG e (d) com o sinal contaminado.

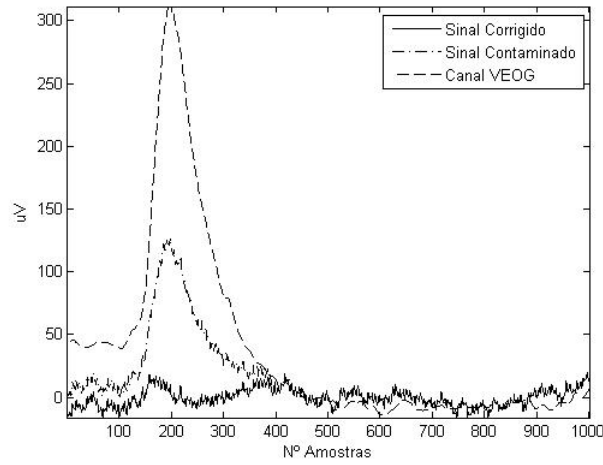


Figura 5.6: Comparação de sinais com canal FP2 depois de aplicado PCA - CORR(10 -11 segundos)

Para o último método, o método de análise canal a canal (SSA - CORR) foi analisado tendo em conta diferentes números de componentes (ver figura 5.8). É mais notória a grande diminuição dos coeficientes de correlação entre o sinal corrigido e o sinal ocular dos canais frontais (primeiros 8 canais) relativamente aos métodos anteriores. Os restantes canais aumentam as variações mas, em média, tendem para nulos (figuras 5.8a, 5.8c e 5.8e). Por outro lado, os coeficientes de correlação do sinal corrigido com o sinal contaminado são semelhantes aos obtidos para os métodos anteriores, havendo um pequeno aumento para os canais a partir do 4.

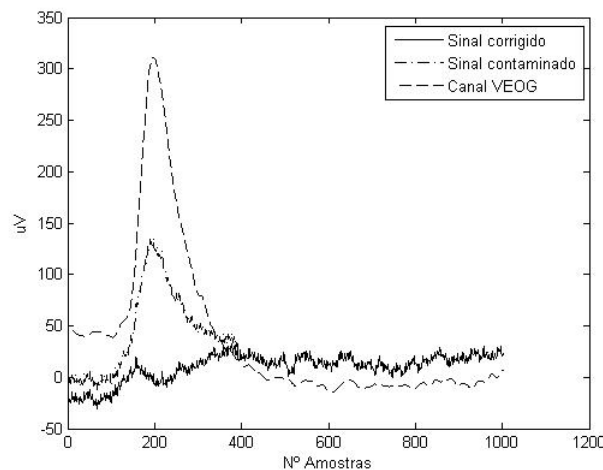


Figura 5.7: Comparação de sinais com canal FPZ depois de aplicado SSA - CORR com 10 componentes (10 -11 segundos)

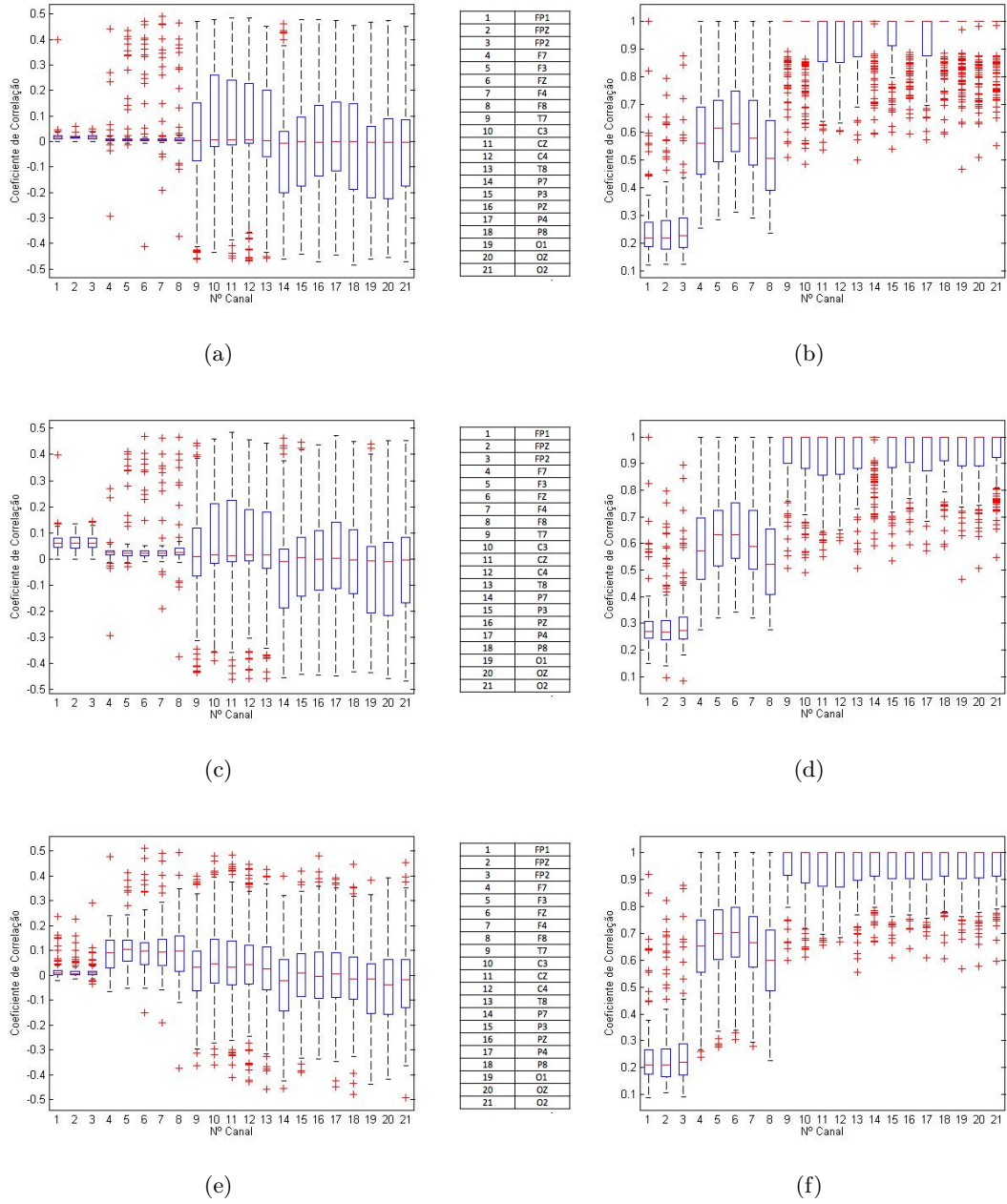


Figura 5.8: Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método (SSA - CORR), calculados nos 400ms de cada pestanejo: Correlação com sinal ocular - (a) M=10 , (c) M=20 e (e) M=50; Correlação com sinal contaminado - (b) M=10 , (d) M=20 e (f) M=50.

5.3.2 Análise tempo - frequência das técnicas implementadas

Aqui é feita uma análise no tempo - frequência dos sinais antes e depois de implementados os diferentes métodos em análise. Para isso, recorre-se à análise espectral dos sinais (espectrograma), e como esta varia ao longo do tempo. Note-se que os sinais analisados estão

compreendidos entre os 10 - 15 segundos.

Da análise da figura 5.9, facilmente se retira que o a maior energia surge com os pestanejos atingindo um máximo de 40dB e uma frequência máxima de 12Hz. Depois de implementado o método regressivo, a energia do sinal diminuiu para um máximo de 20dB, compreendido entre os 0 e os 8Hz. Este método fez assim diminuir um pouco a energia do sinal, concentrando a maior densidade abaixo dos 5/8Hz.

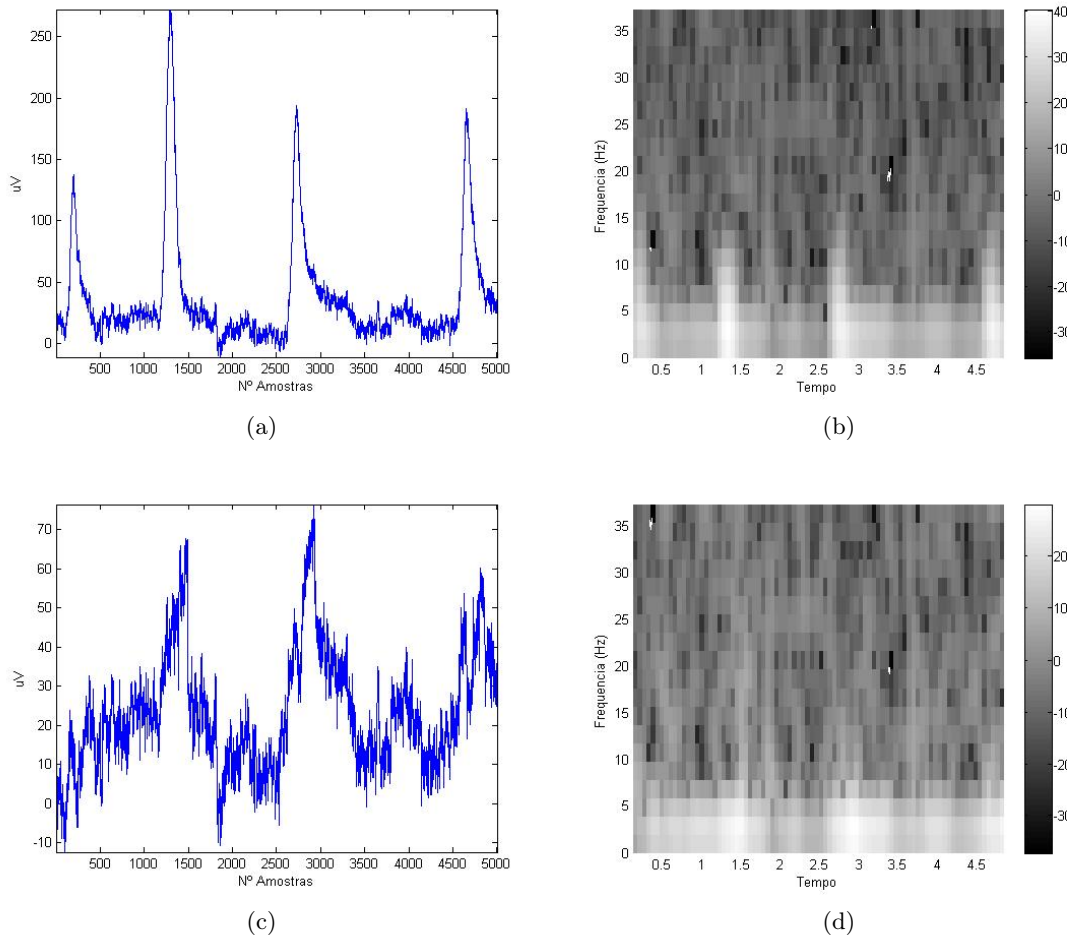


Figura 5.9: Sinal de EEG (canal FP1) entre 10 e 15 segundos: (a) Sinal contaminado, (b) Análise na frequência do sinal contaminado, (c) Sinal corrigido pelo MR, (d) Análise na frequência do sinal corrigido pelo MR.

Para o o método PCA e SSA os resultados são semelhantes. A diferença surge quando em vez de corrigida, a componente é eliminada. Enquanto no método PCA - EL a eliminação faz diminuir a energia do sinal nos intervalos de tempo a que a componente pertencia mas não de uma forma muito significativa (ver figura 5.10), no método SSA - EL a diminuição da energia nos intervalos de tempo a que a componente pertencia é elevada o que provoca uma distorçãom do sinal (ver figura 5.11).

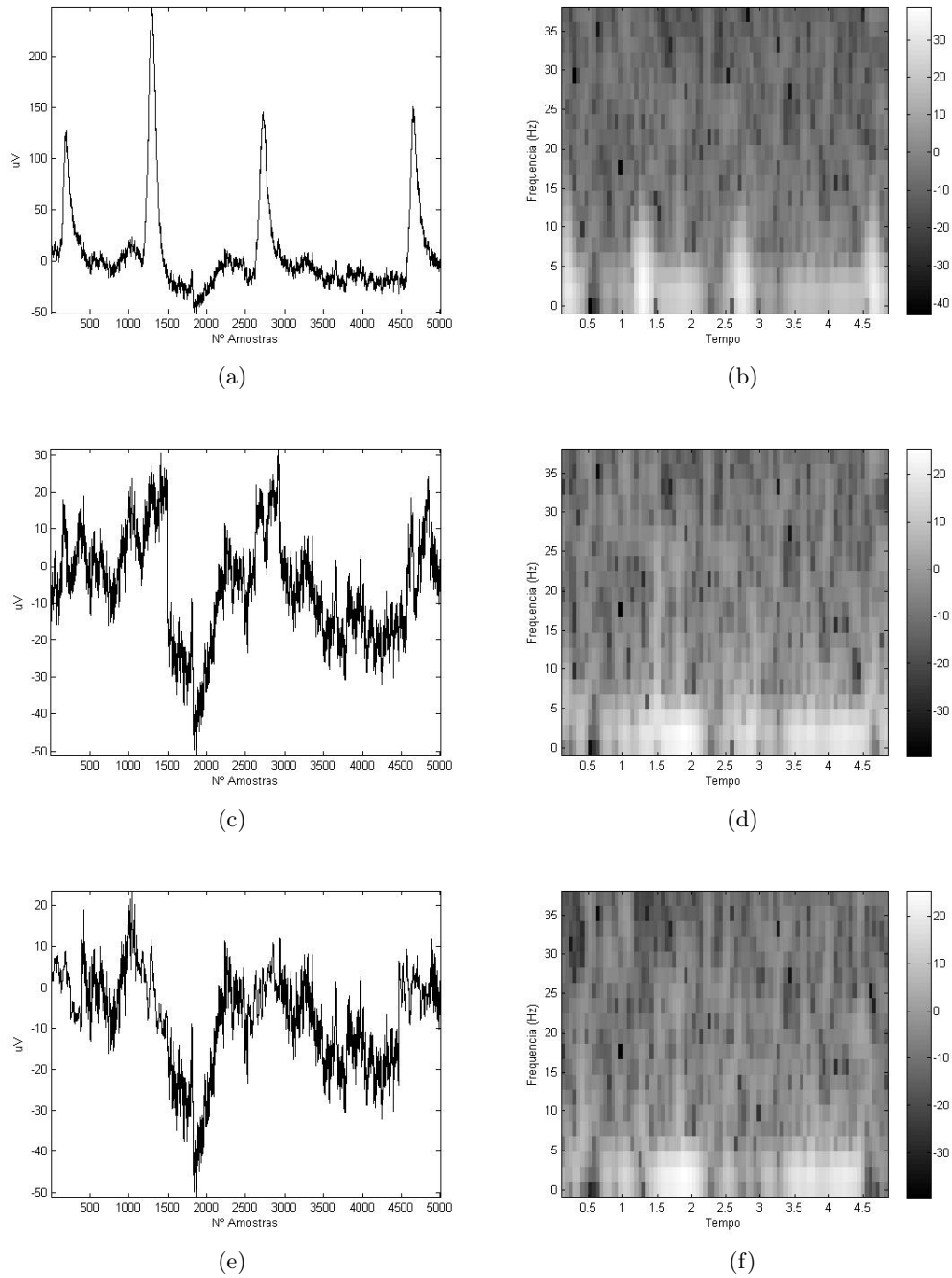
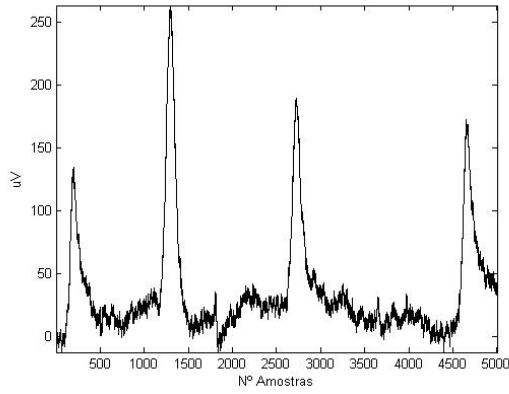
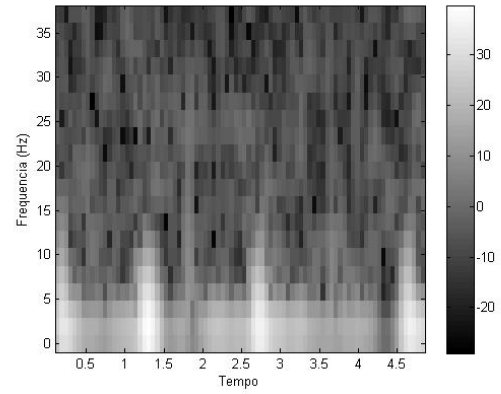


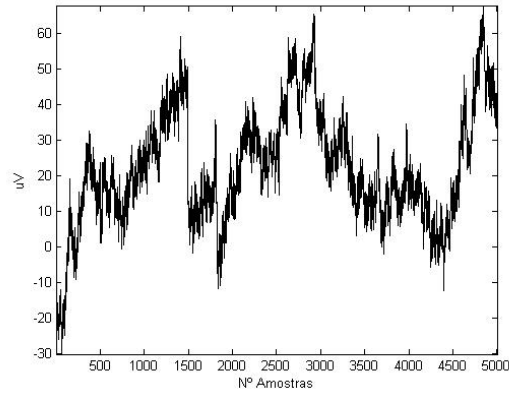
Figura 5.10: Sinal de EEG (canal FP2) entre 10 e 15 segundos: (a) Sinal contaminado, (b) Análise na frequência do sinal contaminado, (c) Sinal corrigido pelo método PCA - CORR, (d) Análise na frequência do sinal corrigido pelo método PCA - CORR, (e) Sinal corrigido pelo método PCA - EL, (f) Análise na frequência do sinal corrigido pelo método PCA - EL.



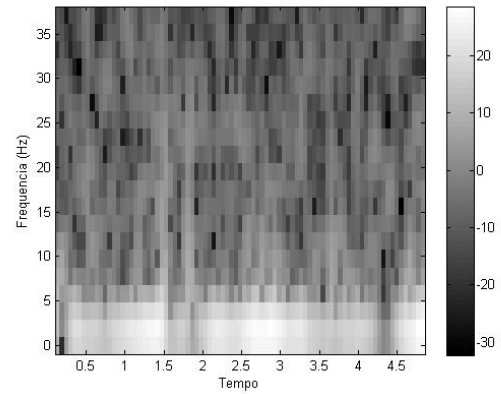
(a)



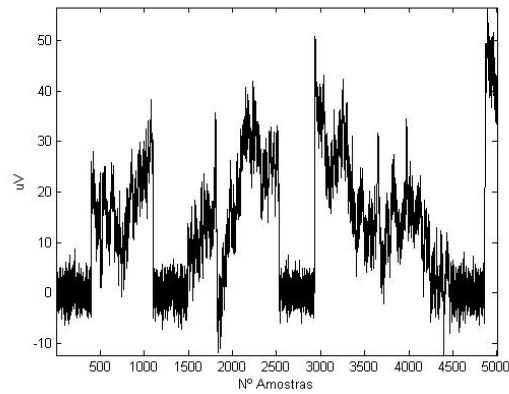
(b)



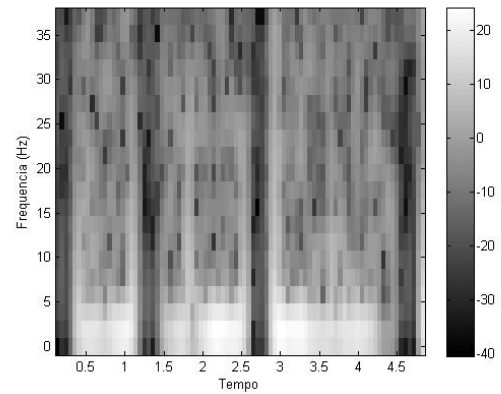
(c)



(d)



(e)

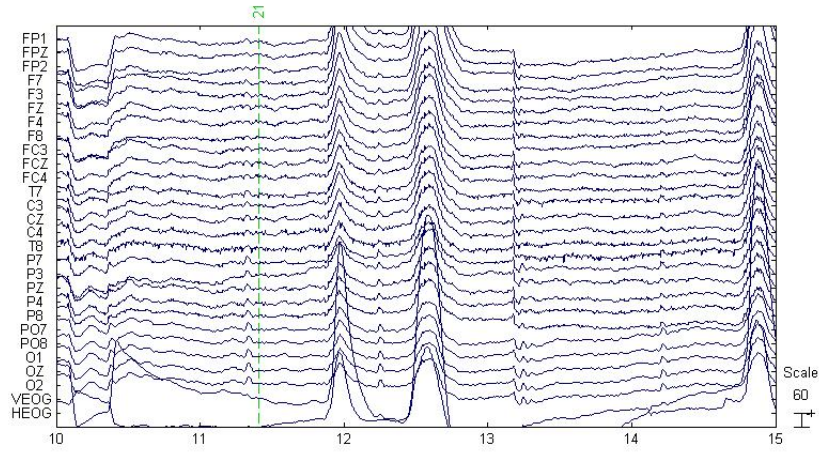


(f)

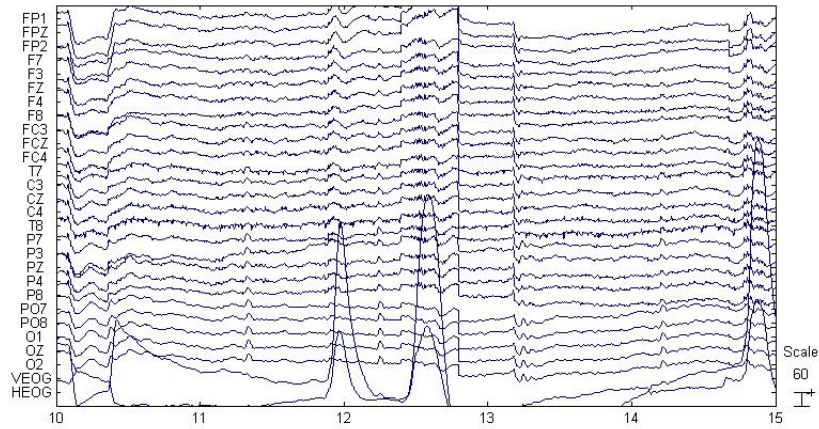
Figura 5.11: Sinal de EEG (canal FPZ) entre 10 e 15 segundos: (a) Sinal contaminado, (b) Análise na frequência do sinal contaminado, (c) Sinal corrigido pelo método SSA - CORR com $M=10$, (d) Análise na frequência do sinal corrigido pelo método SSA - CORR com $M=10$, (e) Sinal corrigido pelo método SSA - EL com $M=10$, (f) Análise na frequência do sinal corrigido pelo método SSA - EL com $M=10$.

5.4 Características do Registo 2

Os sinais do registo 2 apresentam uma duração mais elevada que os do registo 1 (cerca de 25 minutos), e diferem destes na aquisição, também. Estes sinais não foram adquiridos de acordo com o protocolo descrito no capítulo “Artefactos e EEG”. Para além de o número de canais ser maior (28 canais dos quais 2 são oculares), os pestanejos acontecem de uma forma irregular e algumas vezes seguidos, o que faz com que a tarefa de corrigir os artefactos oculares seja dificultada (ver figura 5.12). Para além disto, estão presentes 416 estímulos e foram tidos em conta para o cálculo dos coeficientes de correlação 171 pestanejos.



(a)



(b)

Figura 5.12: (a) Sinal de EEG contaminado e (b) depois de corrigido pelo método PCA - CORR

Numa primeira análise, os coeficientes de correlação entre o sinal ocular VEOG e sinal de EEG são praticamente unitários, com algumas variações. A grande diferença, relativamente

ao sinais anteriores, é os coeficientes de correlação serem elevados em todos os canais, o que significa que todos os canais, de uma forma geral, sofrem alterações com os artefactos oculares existentes(ver figura 5.13). Uma visualização do canal ocular e do canal FP1 pode ser vista na figura 5.14.

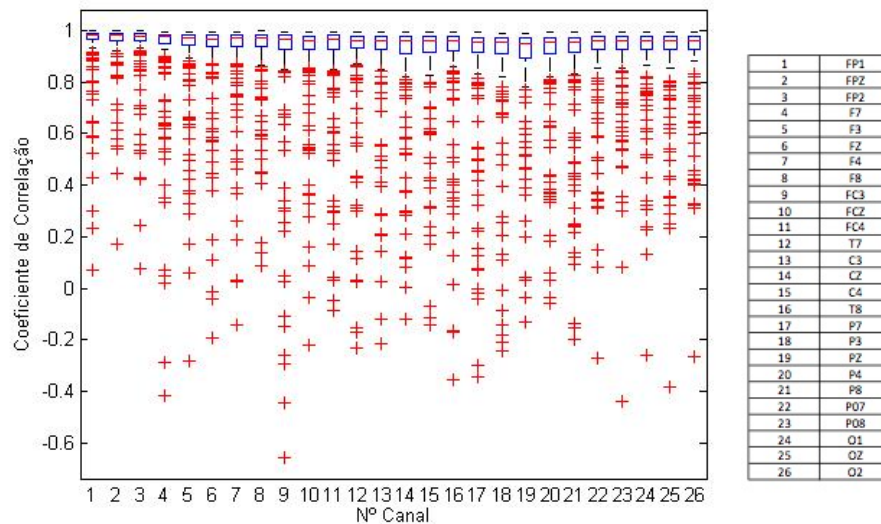


Figura 5.13: Caixa de bigodes dos coeficientes de correlação entre o sinal contaminado e o canal ocular VEOG calculados nos 400ms de cada pestanejo

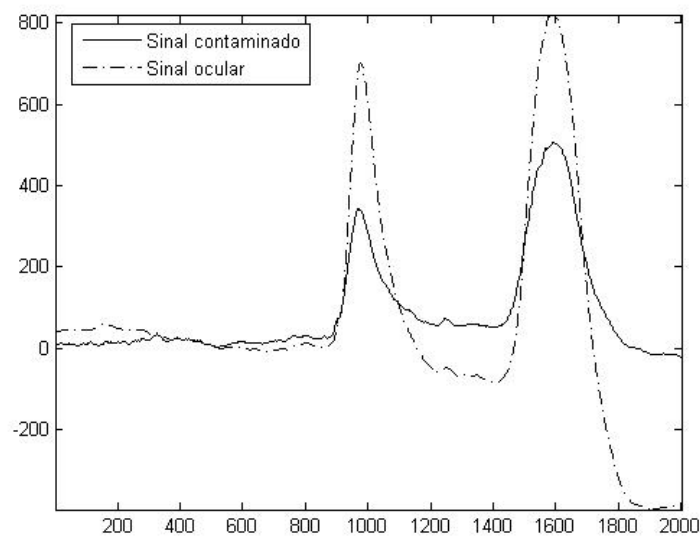


Figura 5.14: Canal FP1 e Canal VEOG (11 -13 segundos)

Como acontecia nos sinais do registo 1, quando se faz a correção dos artefactos oculares com o MR, os coeficientes de correlação entre o sinal corrigido e o sinal ocular passam a ser praticamente nulos (ver figura 5.15a), o que era de esperar uma vez que o artefacto ocular foi removido. A diferença é que aqui existiam artefactos em todos os canais, logo todos os canais apresentam coeficientes de correlação quase nulos. Quanto ao coeficientes de correlação entre o sinal contaminado e o sinal corrigido, apesar de mais elevados que os com o sinal ocular, também são quase nulos (0.2 a 0.3) . Na figura 5.16 pode-se ver em pormenor a correção de dois pestanejos próximos pelo método regressivo.

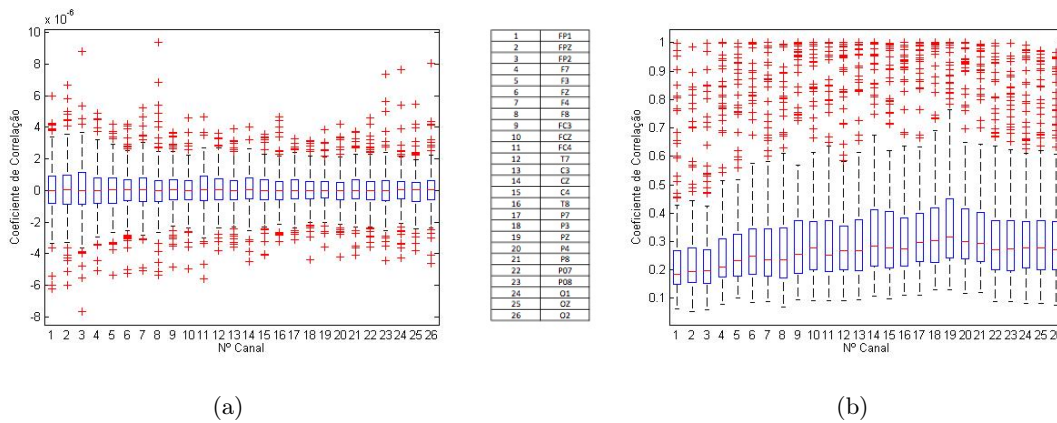


Figura 5.15: Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo MR: (a) com o canal ocular VEOG e (b) com o sinal contaminado, calculados nos 400ms de cada pestanejo

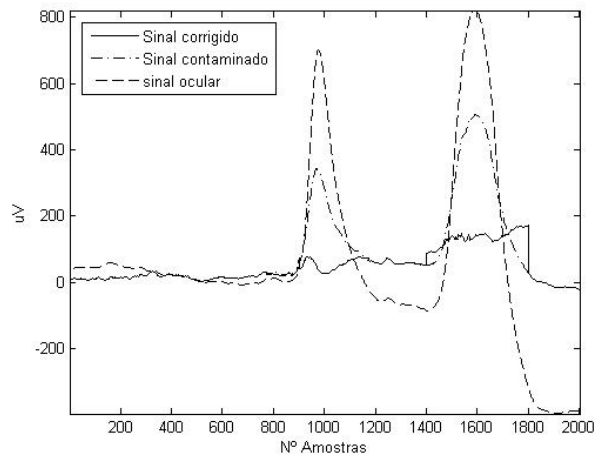
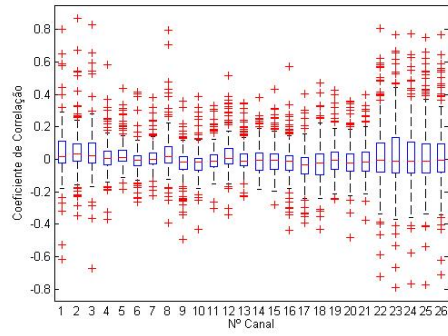


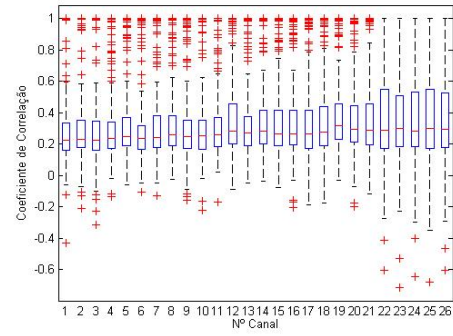
Figura 5.16: Comparação de sinais com canal FP1 depois de aplicado MR(11 -13 segundos)

Com o método (PCA) os resultados são semelhantes aos com os do método MR. São obtidos melhores resultados com a opção de correção de artefacto comparativamente com a eliminação de artefacto.

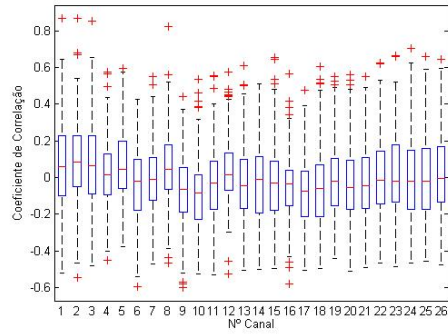


(a)

1	FP1
2	FP2
3	FP2
4	T7
5	F3
6	FZ
7	C4
8	F8
9	FC3
10	FCZ
11	FC4
12	T7
13	C3
14	CZ
15	C4
16	T8
17	P7
18	P3
19	PZ
20	P4
21	P8
22	PO7
23	PO8
24	O1
25	O2
26	O2

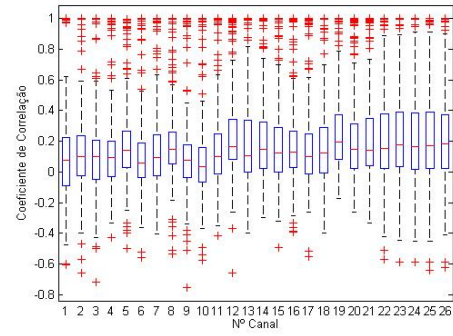


(b)



(c)

1	FP1
2	FP2
3	FP2
4	T7
5	F3
6	FZ
7	C4
8	F8
9	FC3
10	FCZ
11	FC4
12	T7
13	C3
14	CZ
15	C4
16	T8
17	P7
18	P3
19	PZ
20	P4
21	P8
22	PO7
23	PO8
24	O1
25	O2
26	O2



(d)

Figura 5.17: Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método: PCA - CORR: (a) com o canal ocular VEOG e (b) com o sinal contaminado; PCA - EL: (c) com o canal ocular VEOG e (d) com o sinal contaminado.

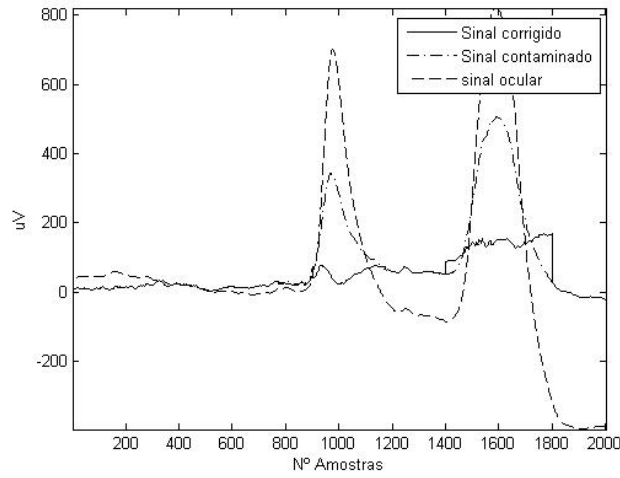


Figura 5.18: Comparação de sinais com canal FP2 depois de aplicado PCA - CORR(11 -13 segundos)

Com o método SSA - CORR os resultados também são semelhantes aos métodos anteriores (ver figura 5.20). Os coeficientes de correlação entre o sinal corrigido e o sinal contaminado indicam que existem alguns canais que não foram corrigidos em alguns pestanejos, isto porque existem coeficientes de correlação unitários. Esta situação é mais notória com o aumento do número de componentes. Os coeficientes que não são unitários, mas são quase, significam que esse canal é pouco afetado com um determinado pestanejo.

Na figura 5.19 pode-se ver em pormenor a correção feita com M=10.

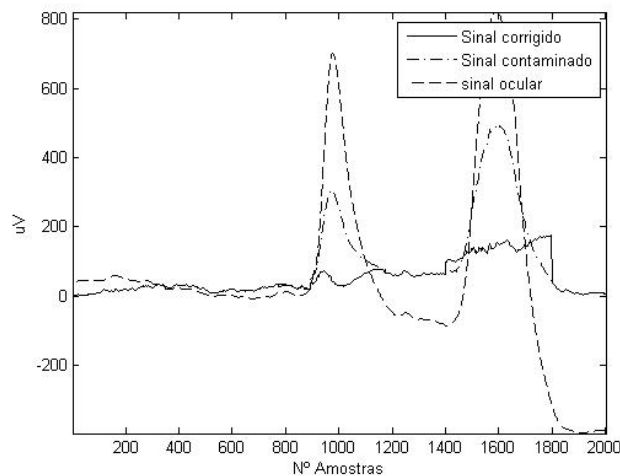
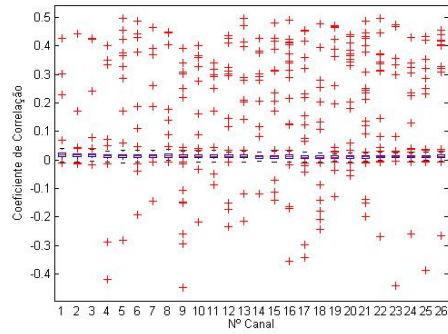
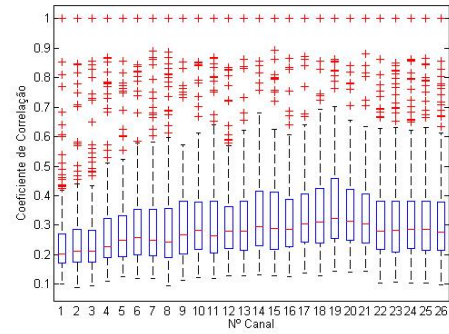


Figura 5.19: Comparação de sinais com canal FPZ depois de aplicado método SSA - CORR com 10 componentes (11 -13 segundos)

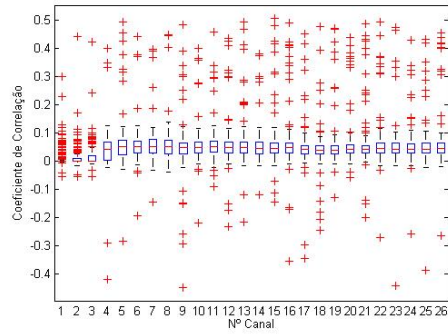


(a)

1	FP1
2	FP2
3	FP2
4	F7
5	F3
6	F2
7	F4
8	F8
9	FC3
10	FC2
11	FC4
12	T7
13	C3
14	C2
15	C4
16	T8
17	P2
18	P3
19	P2
20	P4
21	P8
22	P07
23	P08
24	O1
25	O2
26	O2

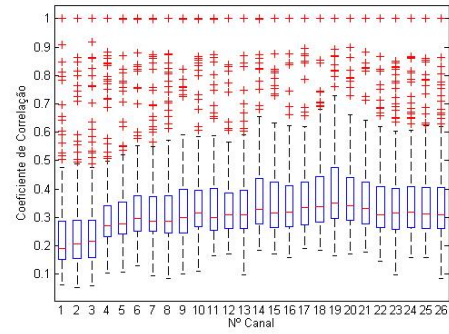


(b)

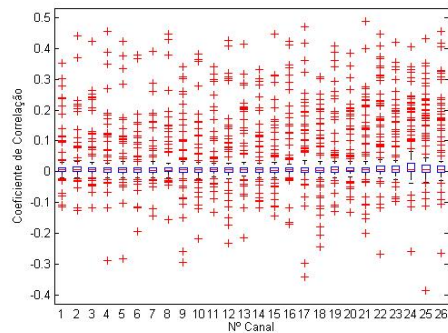


(c)

1	FP1
2	FP2
3	FP2
4	F7
5	F3
6	F2
7	F4
8	F8
9	FC3
10	FC2
11	FC4
12	T7
13	C3
14	C2
15	C4
16	T8
17	P2
18	P3
19	P2
20	P4
21	P8
22	P07
23	P08
24	O1
25	O2
26	O2

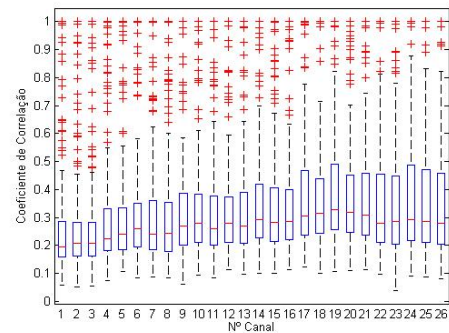


(d)



(e)

1	FP1
2	FP2
3	FP2
4	F7
5	F3
6	F2
7	F4
8	F8
9	FC3
10	FC2
11	FC4
12	T7
13	C3
14	C2
15	C4
16	T8
17	P2
18	P3
19	P2
20	P4
21	P8
22	P07
23	P08
24	O1
25	O2
26	O2



(f)

Figura 5.20: Caixa de bigodes dos coeficientes de correlação do sinal corrigido pelo método SSA - CORR, calculados nos 400ms de cada pestanejo: Correlação com sinal ocular e (a) M=10, (c) M=20 e (e) M=50; Correlação com sinal contaminado e (b) M=10, (d) M=20 e (f) M=50.

Capítulo 6

Conclusões e trabalho futuro

6.1 Discussão geral dos resultados

Este projeto apresentou a criação de um sistema que corrige automaticamente os sub-segmentos, afetados por artefactos oculares. Para isso, recorre, para além da regressão linear simples, a mais dois tipos de técnicas baseadas em valores singulares. Para avaliar os métodos, fizeram-se testes com dois tipos de registos e observaram-se os resultados que foram expostos anteriormente. Desses resultados, várias são as conclusões que se podem retirar. Uma das maiores vantagens da utilização das técnicas em vez de uma regressão linear simples é que, geralmente, estas permitem uma melhor correção do artefacto, isto porque recorrem a componentes e não atuam diretamente no sinal. O aumento da correlação entre o sinal corrigido pelo método PCA e o canal ocular vem contrariar esta vantagem. Contudo, este aumento pode ser justificado com uma das desvantagens indicadas anteriormente, onde dizia que o método PCA não consegue separar completamente os artefactos oculares do sinal de EEG. Para além disso, e tendo em conta que no método SSA também ocorre um ligeiro aumento, pode-se concluir que a regressão linear simples, aplicada diretamente ao sinal, pode estar a retirar informação relevante ao sinal de EEG, daí a correlação praticamente nula.

Com eliminação das componentes relacionadas com o EOG, o PCA é melhor que o SSA. No entanto, ambas ficam mais dependentes do limiar que se escolhe para considerar a componente correlacionada com o VEOG. Optando por correção das componentes, pode-se utilizar limiares mais baixos o que permite obter melhores resultados. A grande vantagem que surge com o método PCA implementado neste trabalho, relativamente a outros métodos que já utilizam esta técnica é que este permite a eliminação/correção das componentes que apresentem os valores próprios acima de um limiar previamente definido, e não apenas eliminação da componente com o valor próprio mais elevado.

Quando se faz uma análise para registos mais complexos de correção de artefactos oculares, conclui-se que também são obtidos bons resultados, o que leva a que estas técnicas podem-se tornar uma grande alternativa para programas como o Neuroscan que sentem dificuldades em corrigir este tipo de registos.

Um dos maiores problemas que resultaram da implementação destas técnicas surge porque recorre-se a uma “colagem” dos trechos de sinal corrigido no sinal original. Esta “colagem” nem sempre resulta bem uma vez que resulta num desvio do sinal. Isto acontece porque os 400ms utilizados para duração do pestanejo, por vezes, tornam-se um intervalo de tempo pequeno. Um aumento do tempo estipulado para duração de um pestanejo podia passar por

alternativa, contudo não é solução uma vez que iriam existir situações de sobreposições de pestanejos.

Como mencionado anteriormente, de um modo geral os resultados obtidos são bastante satisfatórios e, desta forma, surge a possibilidade de idealizar projetos futuros com a implementação destas técnicas como método alternativo à correção de artefactos oculares presentes em sinais de EEG.

6.2 Trabalho futuro

Como trabalho futuro o que mais se destaca é a criação de um interface que não o seja apenas com a finalidade de facilitar a avaliação das técnicas, mas sim uma interface para o utilizador que recorra a estas técnicas e que permita a este corrigir qualquer sinal de EEG contaminado por artefactos oculares. Que permita também ao utilizador considerar válido ou não um pico detetado como sendo artefacto ocular.

Pode ainda ser identificado como trabalho futuro o desenvolvimento de uma solução para os desvio causados pelas “colagens” dos trechos de sinal corrigido. Uma solução poderia passar por não colocar uma duração fixa para todos os pestanejos, sendo esta determinada por software tendo em conta um limiar previamente predefinido. Assim, o início e o fim do pestanejo eram identificados sempre que o sinal ultrapassasse superiormente e inferiormente esse limiar.

Mencionadas as propostas de trabalho futuro, a evolução deste projeto fica apenas limitado à criatividade das pessoas que o possam vir a abraçar.

Apêndices

Apêndice A

Exemplo de uma estrutura criada pela EEGLab

Nome	Valor
setname	'CNT file'
filename	' '
(...)	
comments	[1x56 char]
nbchan	23
trials	1
pnts	345373
srate	1000
xmin	0
xmax	345.3720
times	[]
data	[23x345373 single]
(...)	
chanlocs	[1x23 struct]
urchanlocs	[]
chaninfo	[1x1 struct]
ref	'common'
event	[1x71 struct]
urevent	[1x71 struct]
eventdescription	{ [] [] ' ' }
epoch	[]
epochdescription	
reject	[1x1 struct]
stats	[1x1 struct]
specdata	[]
(...)	
dipfit	[]
history	' '
saved	'no'

Figura A.1: Exemplo de uma estrutura de dados do EEGLab, depois de carregado ficheiro contínuo

Apêndice B

Código Matlab desenvolvido

B.1 Interface gráfica

```
function varargout = teste_15_06(varargin)
%TESTE_15_06 M-file for teste_15_06.fig
%   TESTE_15_06, by itself, creates a new TESTE_15_06 or raises the existing
%   singleton*.
%
%   H = TESTE_15_06 returns the handle to a new TESTE_15_06 or the handle to
%   the existing singleton*.
%
%   TESTE_15_06('Property','Value',...) creates a new TESTE_15_06 using the
%   given property value pairs. Unrecognized properties are passed via
%   varargin to teste_15_06_OpeningFcn. This calling syntax produces a
%   warning when there is an existing singleton*.
%
%   TESTE_15_06('CALLBACK') and TESTE_15_06('CALLBACK', hObject,...) call the
%   local function named CALLBACK in TESTE_15_06.M with the given input
%   arguments.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help teste_15_06
%
% Last Modified by GUIDE v2.5 02-Aug-2011 11:48:35
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @teste_15_06_OpeningFcn, ...
                  'gui_OutputFcn',  @teste_15_06_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```

% End initialization code - DO NOT EDIT

% — Executes just before teste_15_06 is made visible.
function teste_15_06_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)

% Choose default command line output for teste_15_06
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes teste_15_06 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% — Outputs from this function are returned to the command line.
function varargout = teste_15_06_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% — Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
%handles = guidata(hObject);

% create and save method variable
metodo=get(handles.grupo_metodo, 'SelectedObject');

switch get(metodo, 'Tag') % Get Tag of selected object
    case 'met1'
        disp('Foi selecionado metodo 1')
        method=1;

    case 'met2'
        %execute this code when fontsize12_radiobutton is selected
        disp('Foi selecionado metodo 2')
        method=2;

    case 'met3'
        %execute this code when fontsize16_radiobutton is selected
        disp('Foi selecionado metodo 3')
        method=3;

    otherwise
        % Code for when there is no match.
end

% save a selected channel in canais
canal(1)=get(handles.can1, 'Value');
canal(2)=get(handles.can2, 'Value');
canal(3)=get(handles.can3, 'Value');
canal(4)=get(handles.can4, 'Value');

```

```

canal(5)=get(handles.can5,'Value');
canal(6)=get(handles.can6,'Value');
canal(7)=get(handles.can7,'Value');
canal(8)=get(handles.can8,'Value');
canal(9)=get(handles.can9,'Value');
canal(10)=get(handles.can10,'Value');
canal(11)=get(handles.can11,'Value');
canal(12)=get(handles.can12,'Value');
canal(13)=get(handles.can13,'Value');
canal(14)=get(handles.can14,'Value');
canal(15)=get(handles.can15,'Value');
canal(16)=get(handles.can16,'Value');
canal(17)=get(handles.can17,'Value');
canal(18)=get(handles.can18,'Value');
canal(19)=get(handles.can19,'Value');
canal(20)=get(handles.can20,'Value');
canal(21)=get(handles.can21,'Value');
canaiss=find(canal);
disp(canaiss)

% calculates the range
minimo=str2double(get(handles.min,'String'));
disp(minimo)
maximo=str2double(get(handles.max,'String'));
disp(maximo)
intervalo=(minimo*1000+1):(maximo*1000+1);

% Checks the flag chosen
flag=get(handles.corr_rej,'SelectedObject');

switch get(flag,'Tag') % Get Tag of selected object
    case 'rej'
        disp('rejeição')
        flag=1;

    case 'corr'
        %execute this code when fontsize12_radiobutton is selected
        disp('correção')
        flag=2;

    otherwise
        % Code for when there is no match.
end
file=get(handles.dir,'String');

% Calls the main function
main(file,method,canaiss,intervalo,flag);

% — Executes on button press in can2.
function can2_Callback(hObject, eventdata, handles)
% hObject    handle to can2 (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can2

% — Executes on button press in can3.
function can3_Callback(hObject, eventdata, handles)
% hObject    handle to can3 (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can3

```

```

% — Executes on button press in can1.
function can1_Callback(hObject, eventdata, handles)
% hObject    handle to can1 (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of can1

% — Executes during object creation, after setting all properties.
function min_CreateFcn(hObject, eventdata, handles)
% hObject    handle to min (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,' $\leftarrow$ 
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% — Executes during object creation, after setting all properties.
function max_CreateFcn(hObject, eventdata, handles)
% hObject    handle to max (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,' $\leftarrow$ 
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% — Executes during object creation, after setting all properties.
function grupo_metodo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to grupo_metodo (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    empty — handles not created until after all CreateFcns called

% — Executes when selected object is changed in grupo_metodo.
function grupo_metodo_SelectionChangeFcn(hObject, eventdata, handles)
metodo=get(handles.grupo_metodo,'SelectedObject');
if (get(metodo,'Tag')== 'met3')
    set(handles.can1,'enable','off');
    set(handles.can2,'enable','off');
    set(handles.can3,'enable','off');
    set(handles.can4,'enable','off');
    set(handles.can5,'enable','off');
    set(handles.can6,'enable','off');
    set(handles.can7,'enable','off');
    set(handles.can8,'enable','off');
    set(handles.can9,'enable','off');
    set(handles.can10,'enable','off');
    set(handles.can11,'enable','off');
    set(handles.can12,'enable','off');
    set(handles.can13,'enable','off');
    set(handles.can14,'enable','off');
    set(handles.can15,'enable','off');
    set(handles.can16,'enable','off');

```



```

set(handles.can17,'enable','off');
set(handles.can18,'enable','off');
set(handles.can19,'enable','off');
set(handles.can20,'enable','off');
set(handles.can21,'enable','off');
set(handles.rej,'enable','on');
set(handles.corr,'enable','on');
else if get(metodo,'Tag')== 'met1')
    set(handles.can1,'enable','on');
    set(handles.can2,'enable','on');
    set(handles.can3,'enable','on');
    set(handles.can4,'enable','on');
    set(handles.can5,'enable','on');
    set(handles.can6,'enable','on');
    set(handles.can7,'enable','on');
    set(handles.can8,'enable','on');
    set(handles.can9,'enable','on');
    set(handles.can10,'enable','on');
    set(handles.can11,'enable','on');
    set(handles.can12,'enable','on');
    set(handles.can13,'enable','on');
    set(handles.can14,'enable','on');
    set(handles.can15,'enable','on');
    set(handles.can16,'enable','on');
    set(handles.can17,'enable','on');
    set(handles.can18,'enable','on');
    set(handles.can19,'enable','on');
    set(handles.can20,'enable','on');
    set(handles.can21,'enable','on');
    set(handles.rej,'enable','off');
    set(handles.corr,'enable','off');
else
    set(handles.can1,'enable','on');
    set(handles.can2,'enable','on');
    set(handles.can3,'enable','on');
    set(handles.can4,'enable','on');
    set(handles.can5,'enable','on');
    set(handles.can6,'enable','on');
    set(handles.can7,'enable','on');
    set(handles.can8,'enable','on');
    set(handles.can9,'enable','on');
    set(handles.can10,'enable','on');
    set(handles.can11,'enable','on');
    set(handles.can12,'enable','on');
    set(handles.can13,'enable','on');
    set(handles.can14,'enable','on');
    set(handles.can15,'enable','on');
    set(handles.can16,'enable','on');
    set(handles.can17,'enable','on');
    set(handles.can18,'enable','on');
    set(handles.can19,'enable','on');
    set(handles.can20,'enable','on');
    set(handles.can21,'enable','on');
    set(handles.rej,'enable','on');
    set(handles.corr,'enable','on');
end
end

% — Executes on button press in browser.
function browser_Callback(hObject, eventdata, handles)
% hObject    handle to browser (see GCBO)
% eventdata  reserved — to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[temp path]=uigetfile('*.cnt');
file=strcat(path,temp); % get a file

```

```

set(handles.dir,'String',file)

% — Executes during object creation, after setting all properties.
function dir_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    empty – handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'↵
    defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function min_Callback(hObject, eventdata, handles)
% hObject    handle to min (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of min as text
%       str2double(get(hObject,'String')) returns contents of min as a
%       double

function max_Callback(hObject, eventdata, handles)
% hObject    handle to max (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of max as text
%       str2double(get(hObject,'String')) returns contents of max as a double

% — Executes on button press in can4.
function can4_Callback(hObject, eventdata, handles)
% hObject    handle to can4 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can4

% — Executes on button press in can5.
function can5_Callback(hObject, eventdata, handles)
% hObject    handle to can5 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can5

% — Executes on button press in can6.
function can6_Callback(hObject, eventdata, handles)
% hObject    handle to can6 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can6

% — Executes on button press in can7.
function can7_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to can7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can7

% — Executes on button press in can8.
function can8_Callback(hObject, eventdata, handles)
% hObject    handle to can8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can8

% — Executes on button press in can12.
function can12_Callback(hObject, eventdata, handles)
% hObject    handle to can12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can12

% — Executes on button press in can13.
function can13_Callback(hObject, eventdata, handles)
% hObject    handle to can13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can13

% — Executes on button press in can14.
function can14_Callback(hObject, eventdata, handles)
% hObject    handle to can14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can14

% — Executes on button press in can15.
function can15_Callback(hObject, eventdata, handles)
% hObject    handle to can15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can15

% — Executes on button press in can16.
function can16_Callback(hObject, eventdata, handles)
% hObject    handle to can16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can16

% — Executes on button press in can17.
function can17_Callback(hObject, eventdata, handles)
% hObject    handle to can17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hint: get(hObject,'Value') returns toggle state of can17

% — Executes on button press in can18.
function can18_Callback(hObject, eventdata, handles)
% hObject    handle to can18 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can18

% — Executes on button press in can19.
function can19_Callback(hObject, eventdata, handles)
% hObject    handle to can19 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can19

% — Executes on button press in can11.
function can11_Callback(hObject, eventdata, handles)
% hObject    handle to can11 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can11

% — Executes on button press in can10.
function can10_Callback(hObject, eventdata, handles)
% hObject    handle to can10 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can10

% — Executes on button press in can9.
function can9_Callback(hObject, eventdata, handles)
% hObject    handle to can9 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can9

% — Executes on button press in can21.
function can21_Callback(hObject, eventdata, handles)
% hObject    handle to can21 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can21

% — Executes on button press in can20.
function can20_Callback(hObject, eventdata, handles)
% hObject    handle to can20 (see GCBO)
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of can20

```

```
% — Executes on button press in visualizar.
function visualizar_Callback(hObject, eventdata, handles)
% hObject      handle to visualizar (see GCBO)
% eventdata    reserved — to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
file=get(handles.dir, 'String');
EEG = pop_loadcnt(file, 'dataformat', 'int32'); % load file
EEG = eeg_checkset( EEG );
eeg=EEG.data;
% Plot of the EEG signal
eegplot(eeg, 'srate', EEG.srate, 'eloc_file', EEG.chanlocs, 'events', EEG.event)
```

B.2 Detector de picos

```
function [maxtab, mintab]=peakdet(v, delta, x)
%PEAKDET Detect peaks in a vector
% [MAXTAB, MINTAB] = PEAKDET(V, DELTA) finds the local
% maxima and minima ("peaks") in the vector V.
% MAXTAB and MINTAB consists of two columns. Column 1
% contains indices in V, and column 2 the found values.
%
% With [MAXTAB, MINTAB] = PEAKDET(V, DELTA, X) the indices
% in MAXTAB and MINTAB are replaced with the corresponding
% X-values.
%
% A point is considered a maximum peak if it has the maximal
% value, and was preceded (to the left) by a value lower by
% DELTA.

% Eli Billauer, 3.4.05 (Explicitly not copyrighted).
% This function is released to the public domain; Any use is allowed.

maxtab = [];
mintab = [];

v = v(:); % Just in case this wasn't a proper vector

if nargin < 3
    x = (1:length(v))';
else
    x = x(:);
    if length(v) ~= length(x)
        error('Input vectors v and x must have same length');
    end
end

if (length(delta(:)))>1
    error('Input argument DELTA must be a scalar');
end

if delta <= 0
    error('Input argument DELTA must be positive');
end

mn = Inf; mx = -Inf;
mnpos = NaN; mxpos = NaN;

lookformax = 1;

for i=1:length(v)
    this = v(i);
```

```

if this > mx, mx = this; mxpos = x(i); end
if this < mn, mn = this; mnpos = x(i); end

if lookformax
    if this < mx-delta
        maxtab = [maxtab ; mxpos mx];
        mn = this; mnpos = x(i);
        lookformax = 0;
    end
else
    if this > mn+delta
        mintab = [mintab ; mnpos mn];
        mx = this; mxpos = x(i);
        lookformax = 1;
    end
end
end
end

```

B.3 Função principal

```

function []=main(file,method,canais,intervalo,flag)

% load file
EEG = pop_loadcnt(file , 'dataformat', 'int32');
EEG = eeg_checkset( EEG );
eeg=EEG.data;

% Division by methods
switch method
    case 1
        index= peaks(eeg,intervalo); % get eye movements
        blink_dur=200; % Peak duration estimated is 400ms

        for i=1:length(index)
            for j=1:length(canais)
                %%%%%%%%%% the blink analysis %%%%%%%%%%%%%%
                y=eeg(canais(j),index(i)-blink_dur:index(i)+blink_dur);
                x=eeg(22,index(i)-blink_dur:index(i)+blink_dur);

                % test variable
                temp=corrcoef(y,x);
                corr1(j,i)=temp(1,2);

                %Calls the "metodo regressivo"
                Y=metodoregressivo(y,x);

                %test variable
                temp1=corrcoef(Y,x);
                corr_reg(j,i)=temp1(1,2);

                %reconstructs the eeg signal
                eeg(canais(j),index(i)-blink_dur:index(i)+blink_dur)=Y;
            end
        end
        % plot eeg signal
        eegplot(eeg, 'srate', EEG.srate, 'eloc_file', EEG.chanlocs)

    case 2
        index= peaks(eeg,intervalo); % get eye movements
        blink_dur=200; % Peak duration estimated is 400ms

        for i=1:length(index)

```

```

        for j=1:length(canaiss)
            %%%%%%%%%% the blink blink analysis %%%%%%%%%%
            y=eeg(canaiss(j),index(i)-blink_dur:index(i)+blink_dur);
            x=eeg(22,index(i)-blink_dur:index(i)+blink_dur);

            %test variable
            temp=corrcoef(y,x);
            corr2(j,i)=temp(1,2);

            %Calls the "metodo SSA"
            M=10; % variable that defines delay
            [sa,dd]=ssa(y,M,x,flag);

            %test variable
            temp1=corrcoef(sa,x);
            corr_ssa1(j,i)=temp1(1,2);

            %reconstructs the eeg signal
            eeg(canaiss(j),index(i)-blink_dur:index(i)+blink_dur)=sa;
        end
    end
    % Plot eeg signal
    eegplot(eeg, 'srate', EEG.srate, 'eloc_file', EEG.chanlocs)

case 3
    index= peaks(eeg,intervalo); % get eye movements
    blink_dur=200; % Peak duration estimated is 400ms

    for i=1:length(index)
        %%%%%%%%%% the blink blink analysis %%%%%%%%%%
        Z=eeg(1:21,index(i)-blink_dur:index(i)+blink_dur);
        x=eeg(22,index(i)-blink_dur:index(i)+blink_dur);

        %Calls the "metodo PCA"
        [pcas,DD] =pcasimples(Z,x,flag);

        % Test variable
        for j=1:21
            temp1=corrcoef(pcas(j,:),x);
            corr_pca1(j,i)=temp1(1,2);
        end

        %reconstructs the eeg signal
        eeg(1:21,index(i)-blink_dur:index(i)+blink_dur)=pcas;
    end
    % Plot eeg signal
    eegplot(eeg, 'srate', EEG.srate, 'eloc_file', EEG.chanlocs)
end
end

function index=peaks(eeg,intervalo)
    % Calls peak_detect
    [maxtab mintab]=peakdet(eeg(22,intervalo),100,intervalo);

    %save index
    for k=1:length(maxtab(:,1))
        index(k)=maxtab(k,1);
    end
end
end

```

B.4 Método Regressivo

```

function Y=metodoregressivo(y,x)
%Usage: Y =metodoregressivo(y,x)
%
%inputs:
%y-matrix[1xN] (1 channel of EEG - row vector)
%x-matrix [1xN] (VEOG channel - Row vector)
%
%outputs:
%Y-matrix [1xN] (channel of EEG without artefact)

covariancia=cov(x,y);
a=covariancia(1,2)/var(x);

Y=y-a*x;
end

```

B.5 Método PCA

```

function [X,dd] =pcasimples(Z,x,flag)
%Usage: [X,dd] =pcasimples(Z,x)
%Z- matrix MXN (M channels of EEG)
%x- signal VEOG (row vector)- to be compared
%X- without Z artefact and dd the correlation coefficients.
%flag- choose between correction and rejection

THRESHOLD=0.5;
[M,N]=size(Z);
if N~=length(x)
    error('invalid input')
end
% using svd the eigenvalues are ordered.
[U,d,dummy]=svd(Z*Z');
d=diag(d); %vector with eigenvalues

X=U'*Z;
dd=[];

%substitute the "similar"
for k=1:M
    dummy=corrcoef(X(k,:),x);
    if abs(dummy(1,2))>= THRESHOLD
        dd=[dd dummy(1,2)];
        %Eliminate completely
        if flag==1
            X(k,:)=zeros(1,N);
        % Correction
        elseif flag==2
            covariancia=cov(x,X(k,:));
            a=covariancia(1,2)/var(x);
            X(k,:)=X(k,:)-a*x;
        end
    end
end
end
X=U*X;

```

B.6 Método SSA


```

function [z,dd]=ssa(x,M,y,flag)
%Usage: [z,dd]=ssa(x,M,y,flag)
% INPUT:
% x - matrix 1xN (1 channel of EEG) 400ms of blink
% M - embedding dimension
% y - signal VEOG (row vector)- to be compared
% flag - choose between correction and rejection
%
% OUTPUT:
% z- matrix 1xN without eye artifacts
% dd-correlation coefficients.

THRESHOLD=0.5;
[L,N]=size(x);
if L > 1 || M > N
    error('not valid inputs');
end

Z=newdate(x,M); % faz o atraso

% using svd the eigenvalues are ordered.
[U,d,dummy]=svd(Z*Z');
d=diag(d); %vector with eigenvalues
dd=[];

for k=1:M % M is a number of delay
    P=U(:,k)*U(:,k)'; % convert matrix on signal
    X(k,:)=join0(P);

    dummy=corrcoef(X(k,:),y);

    if abs(dummy(1,2))>=THRESHOLD
        dd=[dd dummy(1,2)];
        %Eliminate completely
        if flag==1
            X(k,:)=zeros(1,N);
            % Correction
        elseif flag==2
            covariancia=cov(y,X(k,:));
            a=covariancia(1,2)/var(y);
            %b=mean(x)-mean(y)*a; problemas com a media
            X(k,:)=X(k,:)-a*y;
        end
    end
end
z=sum(X);
end

function X=newdate(x,M)
%usage- X=newdate(x,M)
%embedding each row of x.
%x-matrix data (one time series per row)
%M-embedding dimension ,max delay is M-1
%AMT:2005

X=[];
L=size(x,1); % L=row numbers
%do blocks if x has more than one row
for ik=1:L
    for ic=M:-1:1
        X=[X;x(ik,ic:end-M+ic)];
    end
end
end
end

```

```

function x=join0(X)
%usage x=join0(X)
%averaging the elements in a diagonal
% AMT: 2005

[L,N]=size(X);

x=[];
for k=-L+1:1:N-1
%extracting the elements in a diagonal
    a=mean(diag(X,k));
    x=[x a];
end
end

```

Bibliografia

- [1] Pereira A. Efeitos do neuroticismo nos correlatos psicofisiológicos associados à visualização de imagens fortemente activadoras. Master's thesis, 2009.
- [2] Matlab Central. File exchange. <http://www.mathworks.com/matlabcentral/fileexchange/72-peakdetect-m>. Último acesso em 14/01/2011.
- [3] Van Loan CF. *Matrix Computations*, volume 2. Baltimore:John Hopkins University press, 1989.
- [4] R. J. Croft and R. J. Barry. Removal of ocular artifact from the EEG: a review. *Neurophysiol*, 30:5–19, 2000.
- [5] Departamento de Física da Faculdade de Ciências de Lisboa. Potenciais evocados. <http://kdataserv.fis.fc.ul.pt/~jbatista/biofisica/potenciais/potenciais.htm>. Último acesso em 20/10/2011.
- [6] Merdad Fatourehchi, Ali Bashashati, Rabab K. Ward, and Gary E. Birch. EMG and EOG artifacts in brain computer interface systems: A survey. *Clinical Neurophysiology*, 118:480–494, 2007.
- [7] G. Gratton, M. G. H. Coles, and E. Donchin. A new method for off-line removal of ocular artifacts. *Electroenceph. Clin. Neurophysiol.*, 55:468–484, 1983.
- [8] Nicole Ille, Patrick Berg, and Michael Scherg. Artifact correction of the ongoing EEG using spatial filters based on artifact and brain signal topographies. *Journal of Clinical Neurophysiology*, 19:113–124, 2002.
- [9] Immrama Institute. The international 10-20 system of electrode placement. <http://www.immrama.org/eeg/electrode.html>. Último acesso em 14/09/2011.
- [10] Bell A. J. and Sejnowski T. J. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
- [11] T. P. Jung, M. Westerfield, J. Townsend, E. Courchesne, and T. J. Sejnowski. Removal of eye activity artifacts from visual event-related potential in normal and clinical subjects. *Biological Psychology*, 16:127–147, 2000.
- [12] P. J. Lang, M. M. Bradley, and B. N. Cuthbert. International affective picture system (IAPS): Affective ratings of pictures and instruction manual. Technical report, University of Florida, 2005.

- [13] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis*, volume 4. Wiley, 2006.
- [14] He P., Wilson G., and Russell C. Removal of ocular artifacts from electro-encephalogram by adaptive filtering. *Med. Biol. Eng. Comput*, 42:407–412, 2004.
- [15] Medscape Reference. Somatosensory evoked potentials, general principles. <http://emedicine.medscape.com/article/1139906-overview>. Último acesso em 22/09/2011.
- [16] A. M. Tomé, A. R. Teixeira, N. Figueiredo, I. M. Santos, P. Georgieva, and E. W. Lang. SSA of biomedical signals: A linear invariant systems approach. *Statistics and Its Interface*, 3:345–355, 2010.
- [17] R. Verleger, T. Gasser, and J. MOCKS. Correction of EOG artifacts in event-related potentials of the EEG: Aspects of reliability and validity. *Psychophysiology*, 19:472–480, 1982.
- [18] Lee T. W., Girolami M., and Sejnowski T. J. Independet component analysis using an extended informax algorithm for mixed subguassian and superguassian sources. *Neural Computation*, 11:409–433, 1999.
- [19] J. L. Whitton, F. Lue, and H. Moldofsky. A spectral method for removing eye movement artifacts from the EEG. *Electroenceph. Clin. Neurophysiol.*, 44:735–741, 1978.
- [20] J. C. Woestenburg, M. N. Verbaten, and J. L. Slanger. The removal of the eye-movement artifact from the EEG by regression analysis in the frequency domain. *Clin. Neurophysiol.*, 111:1745–1758, 1983.